Politechnika Wrocławska

Wydział Elektroniki, Fotoniki i Mikrosystemów

KIERUNEK: Automatyka i Robotyka (AIR)

PRACA DYPLOMOWA INŻYNIERSKA

TYTUŁ PRACY:

Wykorzystanie sieci neuronowych do generowania melodii w stylu country

AUTOR:

Marek Wlazło

PROMOTOR:

dr inż. Wojciech Domski

WROCŁAW 2023

Niniejszą pracę dedykuję mojej mamie Barbarze oraz mojemu tacie Zbigniewowi.

Streszczenie

Celem pracy było stworzenie modelu sieci neuronowej, która jest w stanie generować autentyczne melodie w stylu country. W tym celu zdecydowano się na wykorzystanie rekurencyjnych sieci neuronowych. Został skompletowany zbiór danych treningowych składający się z oryginalnych utworów muzycznych w stylu country. Następnie opracowano proces przetwarzania zbioru uczącego. Kolejnym krokiem było dobranie właściwej architektury sztucznych sieci neuronowych. Dalej dobrane zostały parametry związane z procesem nauki. Później wybrano oraz zaimplementowano metody, które w analityczny sposób pozwalają ocenić cechy konkretnego gatunku muzycznego. Ostatnim krokiem było przeanalizowanie zarówno oryginalnych utworów country, jak i melodii wygenerowanych przez wytrenowane modele. W wyniku przeprowadzonych badań odkryto, że model z komórkami LSTM oraz sieć z komórkami GRU są w stanie generować autentyczne utwory w stylu country. Dzięki głębszej analizie generowanych melodii stwierdzono, że model z komórkami LSTM lepiej realizuje zadanie związane z tworzeniem muzyki country.

Summary

The goal of the study was to create a neural network model that is able to generate authentic country-style melodies. For this purpose, it was decided to use recurrent neural networks. A training dataset consisting of original country songs was compiled. Then the process of processing the training database was developed. The next step was to select the appropriate architecture of the artificial neural networks. Next, the parameters related to the learning process were selected. Later, methods were selected and implemented to analytically evaluate the characteristics of a specific musical genre. The final step was to analyze both the original country songs and the melodies generated by the trained models. As a result of the study, it was discovered that the model with LSTM cells and the network with GRU cells were able to generate authentic country-style tunes. Through a deeper analysis of the generated melodies, it was found that the model with LSTM cells better accomplishes the task of creating country music.

Słowa kluczowe: sieci neuronowe, generowanie muzyki country, komórki LSTM, komórki GRU, uczenie głębokie, architektura sieci, warstwy sieci, funkcje aktywacji

Keywords: neural networks, country music generation, LSTM cells, GRU cells, deep learning, network architecture, network layers, activation functions

Spis treści

1 Wstęp **3**
1.1 Teza 5
1.2 Podział pracy 5

2 Sieci neuronowe **7**
2.1 Rekurencyjne sieci neuronowe 9
2.2 Komórki LSTM 9
2.3 Komórki GRU 10

3 Implementacja sztucznej sieci neuronowej **13**
3.1 Dane treningowe 13
3.2 Przetwarzanie danych treningowych 14
3.3 Architektura sieci 16
3.4 Zaimplementowane modele 19
3.5 Uczenie sieci 20

4 Analiza utworów **23**
4.1 Oryginalne utwory country 23
4.2 Utwory wygenerowane przez modele 24

5 Podsumowanie **31**

Załącznik A **33**

Bibilografia **33**

Rozdział 1

Wstęp

Sztuczna inteligencja to rozległa dziedzina nauki, która skupia się na tworzeniu algorytmów i modeli, które są w stanie samodzielnie dokonywać prognoz na podstawie dostępnych danych, podejmować różnego rodzaju decyzje czy też naśladować ludzką inteligencję w inny sposób [13]. Pierwsze badania nad sztuczną inteligencją zaczęły się w latach 50 XX w. W 1950 roku Alan Turing zaproponował test, który pozwalał ocenić czy maszyna jest inteligentna. Test ten polegał na tym, że tzw. „sędzia" prowadzi rozmowę z maszyną oraz człowiekiem, jednak sędzia nie ma pojęcia, który z jego rozmówców jest człowiekiem. Na koniec rozmowy sędzia musiał zdecydować, która ze stron była istotą ludzką [9]. Jeśli maszyna zdołała go przekonać, że jest człowiekiem, uznawano ją za maszynę posiadającą inteligencję. Test Turinga uważany jest za jeden z kamieni milowych w historii badań nad SI. Innym bardzo ważnym wydarzeniem była konferencja Dartmouth, która odbyła się w 1956r [7]. Wydarzenie to uznawane jest za początek sztucznej inteligencji jako dziedziny nauki . Celem konferencji było zapoczątkowanie badań nad maszynami, które będą mogły samodzielnie uczyć się i myśleć.

Obecnie sztuczna inteligencja prężnie się rozwija i można dostrzec ją w coraz większej ilości miejsc. Jej wpływ jest obecny w medycynie [11] – diagnozowanie chorób, transporcie – optymalizacja tras w transporcie publicznym, przemyśle – automatyzacja procesów produkcyjnych, rozrywce – inteligentne systemy gry, a także codziennych technologiach, takich jak asystenci głosowi czy systemy rekomendacji. Dzięki sztucznej inteligencji możemy cieszyć się bardziej zaawansowanymi usługami, które ułatwiają nasze życie. Pomimo wielu zalet, SI posiada również szereg wad o których trzeba pamiętać. Jednym z największych minusów jest niewątpliwie wrażliwość na dane treningowe, które sztuczna inteligencja wykorzystuje w procesie uczenia. Manipulacja danymi czy też obecność jakichkolwiek uprzedzeń w danych treningowych może doprowadzić do niepożądanego zachowania SI lub podejmowania złych decyzji. Przykładem tego typu zjawiska jest chatbot „Tay" [4] firmy Microsoft, który na podstawie interakcji z użytkownikami, miał się uczyć zachowań typowych dla ludzi, jednak internauci zaczęli celowo zasypywać „Tay" treściami rasistowskimi. Doprowadziło to do tego, że chatbot sam zaczął propagować treści rasistowskie i ksenofobiczne.

Coraz większy wpływ sztucznej inteligencji na nasze życie sprawia, że tematem zaczęły się interesować różne organizacje czy też rządy poszczególnych państw. Przykładem tego typu zjawiska jest Parlament Europejski, który od kilku lat rozważał jakie powinno się wprowadzić regulacje dotyczące sztucznej inteligencji, aby przynosiła ona jak najwięcej korzyści i nie łamała europejskich wartości [10]. Unia Europejska podzieliła sztuczną inteligencję w zależności od poziomów ryzyka jakie SI niesie ze sobą. Pierwszy poziom stanowią systemy niedopuszczalnego ryzyka, które będą zakazane. W skład tego typu

systemów wchodzi m.in. rozpoznawanie twarzy czy klasyfikacja ludzi na podstawie ich statusu społeczno-ekonomicznego. Dalej wyróżnione są systemy wysokiego ryzyka czyli takie, które mogą negatywnie wpłynąć na bezpieczeństwo. Oceniane one będą przed wprowadzeniem na rynek oraz przez cały czas pobytu na rynku. Ostatni zbiór to systemy ograniczonego ryzyka. Będą one musiały spełniać minimalne wymogi np. jeśli chodzi o przejrzystość tak, aby użytkownicy mogli świadomie podejmować decyzje. Dodatkowo modele tworzące nowe treści będą musiały informować użytkowników, że treść została wygenerowana przez SI oraz nie będą one mogły generować zakazanych treści [3].

W obrębie sztucznej inteligencji możemy wyróżnić wiele technologii. Jedną z nich jest uczenie maszynowe, które pozwala komputerom uczyć się z dostępnych danych, dzięki czemu mogą samodzielnie podejmować decyzje bez żadnej ingerencji ze strony programisty. Są to algorytmy, które posiadają umiejętność samodoskonalenia się, poprawiania swojej wydajności oraz optymalizacji swojego działania na podstawie analizy danych wejściowych. Wyróżnia się 4 główne rodzaje uczenia maszynowego [8]. W uczeniu nadzorowanym maszynie podawane są zarówno dane wejściowe jak i odpowiadające im prawidłowe dane wyjściowe. Jest to najpopularniejszy model uczenia maszynowego. W uczeniu nienadzorowanym podawane są jedynie dane wejściowe, a algorytm sam musi odnaleźć w nich odpowiednie zależności i struktury. Trzecia metoda to uczenie częściowo nadzorowane, która łączy obie wcześniejsze. Ostatnim rodzajem jest uczenie ze wzmocnieniem, gdzie algorytm podejmuje szereg decyzji. Za każdą decyzję jest oceniany i otrzymuje nagrodę lub karę, dzięki czemu uczy się jakie kroki powinien w danej sytuacji podjąć.

W skład uczenia maszynowego wchodzi uczenie głębokie, które oparte jest na sztucznych sieciach neuronowych. Zadaniem uczenia głębokiego jest rozwiązywanie skomplikowanych problemów takich jak np. rozpoznawanie mowy czy generowanie obrazów na podstawie analizy dużych ilości danych treningowych [12]. W uczeniu głębokim wyróżnia się warstwę wejściową, warstwy ukryte oraz warstwę wyjściową. Każda warstwa składa się z neuronów, które otrzymują dane wejściowe. Następnie neurony przetwarzają w odpowiedni sposób podane informacje i wysyłają je do kolejnych warstw. Tego typu mechanizm pozwala maszynie uczyć się za pomocą własnego przetwarzania danych. Oprócz liczby warstw i neuronów, sieci neuronowe posiadają szereg innych parametrów, które wpływają bezpośrednio na proces nauki oraz jakość generowanych wyników. Przykładem są funkcje aktywacji, które określają wyjście neuronu na podstawie jego wag i aktualnych danych wejściowych.

Jednym z najpopularniejszych zastosowań sieci neuronowych jest predykcja różnego rodzaju wydarzeń takich jak np. ceny na giełdzie czy też wyniki wyborów. Predykcja to proces, który na podstawie danych historycznych oraz aktualnie dostępnych informacji wyszukuje pewne zależności, cechy dzięki czemu jest w stanie przewidzieć przyszłe zdarzenia lub wyniki.

Przykładem predykcji może być proces generowania melodii country. Polega on na przewidywaniu kolejnych nut i akordów, tak aby całość jak najbardziej przypominała utwory w stylu country. Jest to narzędzie, które może znaleźć wiele zastosowań, począwszy od branży filmowej, poprzez gry wideo, aż po reklamy. Możliwe że tego typu model będzie stanowić istotne wsparcie w procesie tworzenia muzyki, inspirując twórców do komponowania oryginalnych utworów.

Celem pracy jest stworzenie modelu generatywnego, który na podstawie danych wejściowych w postaci sekwencji nut i akordów, będzie w stanie przewidzieć sekwencję wyjściową, tak aby całość brzmiała w stylu country. W celu realizacji projektu zebrano odpowiedni zbiór danych treningowych, który został użyty w procesie uczenia. Podczas budowy modelu wykorzystana została rekurencyjna sieć neuronowa.

1.1 Teza

Możliwe jest zbudowanie modelu generatywnego zdolnego do generowania autentycznych melodii w stylu country na podstawie zadanej sekwencji wejściowej w postaci nut i akordów.

1.2 Podział pracy

W drugim rozdziale znajduje się ogólny opis zarówno budowy jak i działania sztucznych neuronów oraz sieci neuronowych. W bardziej szczegółowy sposób opisane są rekurencyjne sieci neuronowe mianowicie ich budowa, zastosowanie, a także wady i zalety. Analogiczne opisy zostały stworzone dla komórek LSTM i GRU.

Kolejny rozdział traktuje o sztucznych sieciach neuronowych, które zostały zaimplementowane na potrzeby projektu. Zawiera on szereg opisów począwszy od krótkiego wstępu, przez sekcje dotyczące zbierania i przetwarzania danych treningowych, aż po opis użytych architektur sieci neuronowych. Omówione są w nim szczegóły związane z wyborem funkcji aktywacji czy technik regularyzacji. Dodatkowo zawiera on diagramy oraz tabele, które pokazują ułożenie warstw, kierunek przepływu informacji oraz szczegółowe parametry dotyczące użytych warstw w zaimplementowanych modelach. Ostatnia sekcja rozdziału poświęcona jest procesowi uczenia modelu, wyborze funkcji straty oraz optymalizatora.

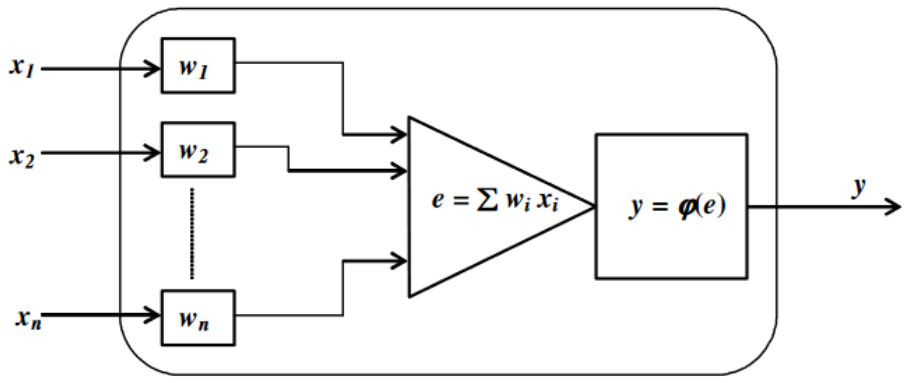
Kolejny rozdział opisuje analizę zarówno oryginalnych utworów country, jak i tych wygenerowanych przez oba modele. Na początku opisane są różne techniki, które zdecydowano się użyć do zbadania wyżej wymienionych melodii. Dalej zawarte są wyniki analiz dla oryginalnych kompozycji. Z wyników tych sformułowane są wnioski, które stanowią wyznacznik cech gatunku country oraz są punktem wyjścia do oceny jakości muzyki generowanej przez wytrenowane sieci. Dalsza analiza dotyczy utworów wygenerowanych przez modele. Poddane są one analizie, głównie w celu sprawdzenia czy zawierają pożądane cechy melodii country. W dalszej części rozdziału umieszczony jest opis dotyczący różnic pomiędzy utworami generowanymi przez oba modele.

Ostatni rozdział stanowi krótkie podsumowanie pracy. Na początku występuje opis rzeczy, które zostały zrealizowane. Dalej opisane są wnioski płynące z analizy utworów wygenerowanych przez oba modele. Na końcu autor wskazuje kilka kierunków związanych z możliwymi dalszymi badaniami.

Rozdział 2

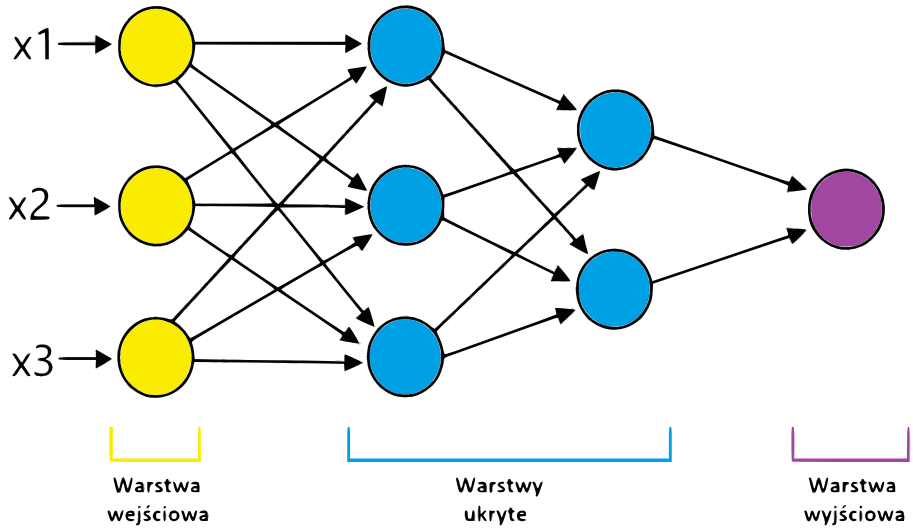
Sieci neuronowe

Sztuczna sieć neuronowa jest narzędziem, które pozwala rozwiązywać problemy z zakresu sztucznej inteligencji oraz uczenia maszynowego. Sieci neuronowe są wzorowane na budowie i działaniu ludzkiego mózgu [6]. Zarówno w sieciach neuronowych jak i w biologicznym mózgu, podstawowymi jednostkami obliczeniowymi są neurony. Budowa sztucznego neuronu została pokazana na rysunku 2.1 Jak można zauważyć neuron składa się z kilku podstawowych elementów. Poprzez wejścia $(x_1, x_2...x_n)$ neuron otrzymuje sygnały wejściowe, które mogą reprezentować różne cechy ważne z punktu widzenia postawionego zadania. Każde wejście ma przypisaną odpowiednią wagę $(w_1, w_2...w_n)$. Wagi określają, jak istotne są poszczególne wejścia dla danego neuronu. Dostosowywanie wag neuronów odbywa się w trakcie procesu uczenia sieci neuronowej, a w szczególności podczas propagacji wstecznej (z ang. *backpropagation*). Po odpowiednim przeskalowaniu danych wejściowych przez odpowiadające im wagi, dochodzi do ich zsumowania w sumatorze ($e = \sum_{i=1}^{n} w_i \cdot x_i$). Daje to możliwość neuronowi przekształcenia wszystkich wejść w jedną wartość. Wynik z sumatora jest przekazywany do funkcji aktywacji ($y = \phi(e)$), która decyduje czy neuron ma być aktywowany na podstawie swoich wejść oraz jaka część informacji znajdująca się w sumatorze ma zostać przesłana dalej, jako wyjście neuronu. Funkcja ta często jest nieliniowa, co pozwala na wprowadzenie nieliniowych relacji pomiędzy wejściami i wyjściami.



Rysunek 2.1 Budowa sztucznego neuronu [15]

Sztuczne sieci neuronowe zbudowane są z warstw, których może być od kilku do kilkunastu. Warstwy zbudowane są ze sztucznych neuronów [14]. Każda sieć posiada jedną warstwę wejściową i wyjściową, dodatkowo występuje co najmniej jedna warstwa ukryta. Przykładową budowę sieci neuronowej można zobaczyć na rysunku 2.2 Warstwa wejściowa jest pierwszą warstwą w sieci. Jej zadaniem jest przyjmowanie danych wejściowych i przekazywanie ich do warstwy ukrytej. Neurony w warstwie wejściowej reprezentują określone cechy, które są ściśle związane z problemem, który ma zostać rozwiązany przez daną sieć neuronową. Warstwy ukryte występują pomiędzy warstwą wejściową, a warstwą wyjściową. Odpowiadają za przetwarzanie danych wejściowych w sposób, który pozwoli na wyodrębnienie pożądanych cech. Każda warstwa ukryta po wykonaniu obliczeń, przekazuje wyniki do kolejnej warstwy ukrytej lub do warstwy wyjściowej. Im większa liczba warstw ukrytych, tym sieć wykazuje większą zdolność do nauki bardziej skomplikowanych wzorców. Przekłada się to bezpośrednio na zwiększone zapotrzebowanie na moc obliczeniową oraz ilość danych wejściowych. Rośnie również ryzyko przeuczenia się sieci. Warstwa wyjściowa jest ostatnią warstwą. Odpowiedzialna jest za generowanie wyników przetwarzania. Neurony w warstwie wyjściowej reprezentują prognozowany wynik dla postawionego zadania. W przypadku generowania muzyki każdy neuron może reprezentować unikalną nutę lub akord, a jego wyjście prawdopodobieństwo wystąpienia danej nuty/akordu.



Rysunek 2.2 Budowa sieci neuronowej

Istnieje wiele różnych architektur sieci neuronowych, z których każda została zaprojektowana w celu rozwiązania określonych problemów w dziedzinie uczenia maszynowego oraz sztucznej inteligencji. Do popularnych i często używanych rodzajów sieci neuronowych można zaliczyć

- konwolucyjne sieci neuronowe – wykorzystywane są głównie do zadań związanych z przetwarzaniem obrazów lub analizą wizualną,
- rekurencyjne sieci neuronowe – przeznaczone są do analizy sekwencji, takich jak analiza mowy czy prognozowanie szeregów czasowych,

- gęsta sieć neuronowa – wykorzystywana w wielu zadaniach, takich jak klasyfikacja czy regresja,
- sieci generatywne GAN – przeznaczone do generowania nowych obrazów, danych i tekstów.

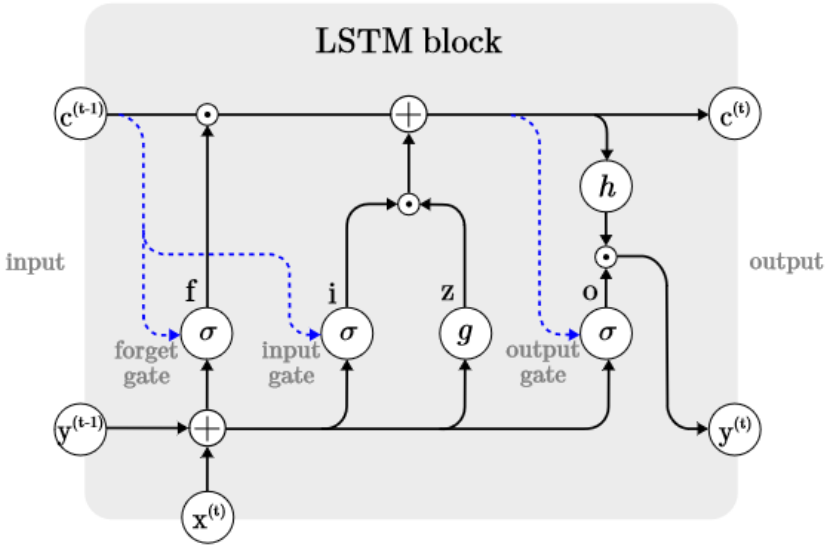
2.1 Rekurencyjne sieci neuronowe

Rekurencyjnie sieci neuronowe (z ang. *Recurrent Neural Network (RNN)*) są jedną z najpopularniejszych architektur sieci neuronowych używanych do rozwiązywania problemów z zakresu uczenia maszynowego, sztucznej inteligencji i przetwarzania sekwencji danych. Główna cecha, która pozwala odróżnić sieci rekurencyjne od standardowych sieci neuronowych, to zdolność do przetwarzania sekwencji danych i zachowywania informacji o poprzednich krokach czasowych. Tradycyjne sieci neuronowe przyjmują dane wejściowe i generują wyniki bez zachowania pamięci o wcześniej przetworzonych danych, podczas gdy sieci rekurencyjne posiadają wewnętrzny stan, który jest przekazywany między kolejnymi krokami czasowymi. W przypadku sieci rekurencyjnych, informacje są przetwarzane krok po kroku, a każdy krok czasowy bierze pod uwagę zarówno bieżące dane wejściowe, jak i stan wewnętrzny wynikający z poprzednich obliczeń. Sprawia to, że sieci rekurencyjne posiadają swego rodzaju sprzężenia zwrotne, czyli połączenia pomiędzy dalszymi warstwami sieci, a bliższymi warstwami ukrytymi. Tego typu architektura pozwala na wykonywanie znacznie bardziej złożonych obliczeń niż w przypadku tradycyjnych sieci. Zdolność do zapamiętywania kontekstu czyni sieci rekurencyjne bardzo przydatnymi w zadaniach, które związane są z szeroko pojętą analizą sekwencji danych, takich jak przetwarzanie języka naturalnego, rozpoznawanie mowy czy też prognozowanie różnego rodzaju zmiennych ewoluujących w czasie np. przewidywanie cen akcji spółek na giełdzie. Sieci RNN posiadają naturalnie wady. Pierwsza z nich dotyczy czasu wymaganego na wyuczenie modelu. W celu uzyskania pożądanych wyników, nierzadko trzeba trenować model przez długi czas. Wraz ze wzrostem długości czasu potrzebnego do wytrenowania sieci, znacząco rosną również wymagania odnośnie zasobów obliczeniowych. Dotyczy to szczególnie głębokich architektur oraz przypadków posiadania dużego zbioru danych wejściowych. Innym problemem z którym borykają się sieci rekurencyjne jest przetrenowanie modelu, które charakteryzuje się niestabilnością uczenia. Problem ten określany jest powszechnie jako eksplodujący gradient. Istnieją różne sposoby radzenia sobie z tego typu problemem, jak np. regularyzacja lub ograniczenie wartości gradientu. Jeszcze innym problemem jest krótka pamięć sieci RNN, która doprowadza do tego, że po pewnym czasie model nie zawiera prawie żadnych informacji pochodzących z początkowych wejść, co znacząco utrudnia pracę na długich sekwencjach danych. Z kolei ten problem określa się mianem zanikającego gradientu. Owe zanikanie odbywa się wykładniczo wraz z czasem. Sieci RNN są modelami sekwencyjnymi, co znacząco ogranicza możliwość do równoległego przetwarzania danych. Prowadzi to do pogorszenia wydajności sieci w przypadku dużych sekwencji danych.

2.2 Komórki LSTM

Komórki LSTM (z ang. *Long Short-Term Memory*) są jedną z najpopularniejszych odmian komórek używanych do budowy rekurencyjnych sieci neuronowych. Główna idea, która przyświecała powstaniu LSTM związana była z potrzebą uzyskania zdolności do zapamiętywania zależności długoterminowych. W tym celu została zaprojektowana nowa archi-

tektura sztucznego neuronu, którą pokazano na rysunku 2.3. Jak można zauważyć zostały wprowadzone specjalne bramki (z ang. *gate*), które odpowiedzialne są za przepływ informacji w neuronie. Bramka zapominania (z ang. *forget gate*) określa, które informacje z poprzednich warstw należy zachować i które trzeba odrzucić. Jej rolą jest eliminowanie nieistotnych danych z przeszłości co ma duże znaczenie w przypadku zadań dotyczących długich sekwencji. Bramka wejściowa (z ang. *input gate*) decyduje, które nowe informacje mają zostać dodane do aktualnego stanu neuronu. Odpowiada za wyodrębnienie najważniejszych informacji z nowych danych i dodawanie ich do stanu komórki. Ostatnią bramką jest bramka wyjścia (z ang. *output gate*), której zadaniem jest określenie wyjścia neuronu. Dzięki tym bramkom, komórki LSTM zyskały pamięć długoterminową, co pozwoliło w znacznym stopniu rozwiązać problem zanikającego gradientu, który występował w sieciach RNN. Dodanie pamięci długoterminowej odbyło się kosztem wzrostu złożoności obliczeniowej, co oznacza m.in. dłuższy czas trenowania modelu.

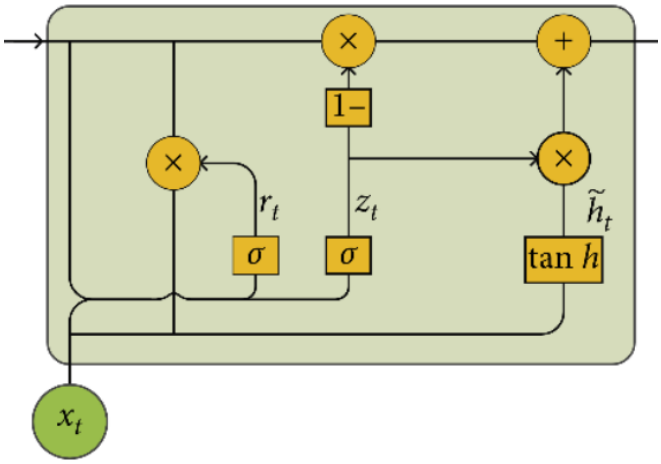


Rysunek 2.3 Architektura neuronu LSTM [5]

2.3 Komórki GRU

Komórki GRU (z ang. *Gated Recurrent Unit*) są kolejnym rodzajem sztucznych neuronów używanych do budowy rekurencyjnych sieci neuronowych. Stanowią one pewien kompromis pomiędzy zwykłymi sieciami RNN, a znaczniej zaawansowanymi komórkami LSTM. Podobnie jak w przypadku LSTM, GRU zostało stworzone z myślą o wyeliminowaniu problemu zanikającego gradientu, z którym borykają się zwykłe komórki RNN. W tym celu do neuronu zostały dodane dwie bramki, które sterują przepływem informacji w sieci. Architekturę neuronu GRU pokazano na rysunku 2.4. Bramka resetu (z ang. *reset gate*) odpowiedzialna jest za określenie, które informacje z wcześniejszych warstw powinny zostać uwzględnione w bieżącym kroku czasowym, a które zapomniane. Bramka aktualizacji (z ang. *update gate*) decyduje, w jaki sposób nowe informacje zostaną dodane lub zaktualizowane w stanie ukrytym. Dodanie tych, dwóch bramek pozwoliło rozwiązać problem zanikającego gradientu (z ang. *vanishing gradient problem*), dzięki czemu możliwe jest znacznie efektywniejsze uczenie modelu różnego rodzaju długoterminowych zależności

w danych. Komórki GRU oraz LSTM wykorzystują wiele podobnych rozwiązań natomiast posiadają również kilka istotnych różnic. Komórka GRU wykorzystuje jedynie 2 bramki do kontroli przepływu informacji, co sprawia, że jej budowa jest mniej złożona. Dodatkowo nie posiada ona oddzielnej komórki pamięci, w której przechowywane są długoterminowe zależności, tak jak ma to miejsce w przypadku LSTM. Ze względu na prostszą architekturę i mniejszą ilość parametrów, GRU jest bardziej wydajne obliczeniowo niż LSTM. Przekłada się to bezpośrednio na skrócony czas, który trzeba poświęcić na uczenie modelu. GRU znajduje zastosowanie w wielu dziedzinach związanych głównie z prognozowaniem i planowaniem, przetwarzaniem języka naturalnego czy analizą czasowych serii danych.



Rysunek 2.4 Architektura neuronu GRU [2]

Rozdział 3

Implementacja sztucznej sieci neuronowej

W celu stworzenia narzędzia zdolnego do generowania melodii w stylu country, wykorzystano różne odmiany sztucznych komórek takich jak LSTM czy GRU. Głównym powodem takiego wyboru jest duża efektywność wyżej wymienionych komórek w obszarze generowania sekwencji, co również obejmuje proces tworzenia melodii muzycznych. Rekurencyjne sieci neuronowe posiadają zdolność do przechowywania informacji z poprzednich kroków czasowych, dzięki czemu mogą one uwzględnić kontekst podczas generowania kolejnych dźwięków danego stylu muzycznego. Możliwość zapamiętywania odgrywa kluczową rolę w tworzeniu spójnych sekwencji dźwięków, ponieważ umożliwia modelowi zrozumienie harmonii oraz tworzenie powiązań pomiędzy nutami i akordami.

3.1 Dane treningowe

W ramach realizacji projektu została stworzona baza treningowa, która stanowi jeden z kluczowych elementów podczas tworzenia sieci neuronowych. Zbiór utworów treningowych powinien być wystarczająco duży. Zbyt mała ilość danych może prowadzić do poważnych problemów podczas trenowania sieci. Pierwszy problem jest związany z niską jakością modelu. Mała baza może sprawić, że sieć neuronowa będzie błędnie wyuczona, co skutkuje znacznie obniżoną jakością predykcji. Dodatkowo może prowadzić to do zjawiska „overfittingu" czyli sytuacji, w której model dobrze radzi sobie z danymi treningowymi natomiast nie jest w stanie poradzić sobie z danymi walidacyjnymi. Innymi słowy oznacza to, że sieć neuronowa traci zdolność do generalizacji. Oprócz samej ilości danych należy zadbać o ich reprezentatywność, która odnosi się do tego czy zebrane dane wystarczająco dobrze odzwierciedlają zróżnicowanie danych, które wytrenowany model będzie musiał przewidywać. Baza treningowa powinna być możliwie mocno zróżnicowana, jednocześnie dbając o to, żeby dane treningowe nie straciły pożądanych wzorców.

W pierwszym kroku udało się zebrać utwory, które cechują się najbardziej autentycznym brzemieniem w stylu country. Starannie wyselekcjonowane kompozycje muzyczne, emanujące elementami charakterystycznymi dla gatunku country stanowią podstawę głównego zbioru treningowego. Pozwalają one na stworzenie realistycznego obrazu stylu muzycznego, który zawiera wiarygodne wzorce potrzebne do nauki modelu. Podzbiór ten liczy około 950 utworów w formacie MIDI. W celu wprowadzenia różnorodności w bazie treningowej został stworzony drugi podzbiór, który składa się głównie z kompozycji zawierających domieszki innych stylów muzycznych np. contry rock czy country pop. Liczy

on około 450 utworów, również w formacie MIDI. Skompletowanie drugiego podzbioru cechującego się różnorodnością ma na celu wprowadzenie większej elastyczności modelu oraz umożliwienie generowania bardziej kreatywnych kompozycji. W wyniku połączenia tych dwóch podzbiorów, powstała baza próbek uczących, która z jednej strony pozwala na naukę charakterystycznych cech gatunku country, natomiast z drugiej strony wprowadza pewną elastyczność w modelu, dzięki której sieć neuronowa może generować bardziej innowacyjne utwory. To zrównoważone podejście do tworzenia bazy treningowej umożliwia modelowi lepsze zrozumienie zarówno koncepcji gatunku, jak i eksperymentowanie z nim. Wszystkie kompozycje wykorzystane do utworzenia bazy treningowej, zostały pobrane ze stron internetowych oferujących bezpłatne pliki w formacie MIDI [1].

3.2 Przetwarzanie danych treningowych

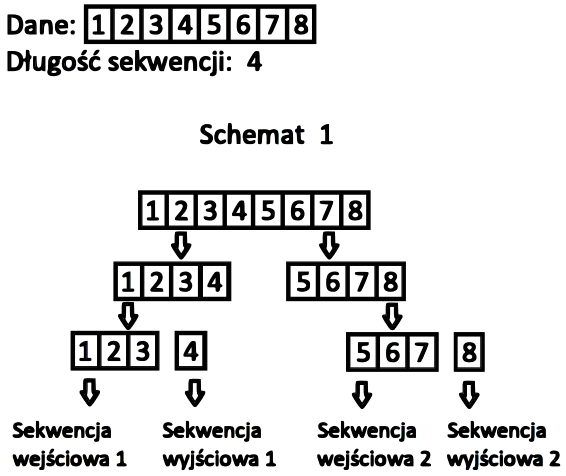
Prawidłowe przygotowanie danych treningowych jest jednym z kluczowych aspektów podczas pracy nad stworzeniem sieci neuronowej. Jest to proces, od którego zależą najważniejsze charakterystyki modelu takie jak wydajność czy skuteczność. Ogólny schemat przetwarzania danych treningowych został zaprezentowany na rysunku 3.3.

Na początku ważne jest opracowanie modelu danych, tzn. określenie, co jest potrzebne do treningu sieci neuronowej. W przypadku tego projektu są to informacje o dźwiękach. Pierwszym krokiem w przetwarzaniu danych jest wydobycie wszystkich informacji o nutach i akordach występujących w utworzonej bazie treningowej. W tym przypadku skupiono się tylko na jednym i jednocześnie najpopularniejszym instrumencie występującym w muzyce country czyli gitarze akustycznej. Wszystkie pozyskane informacje w postaci nut i akordów dotyczą jedynie gitary akustycznej. Związane jest to z tym, że większa liczba instrumentów znacząco powiększyłaby zbiór danych treningowych. To z kolei wiąże się ze znacznym wzrostem czasu potrzebnego na uczenie modelu. Pozyskane informacje o akordach i nutach stanowią fundament danych, który po odpowiednim przetworzeniu będzie użyty w procesie nauki modelu.

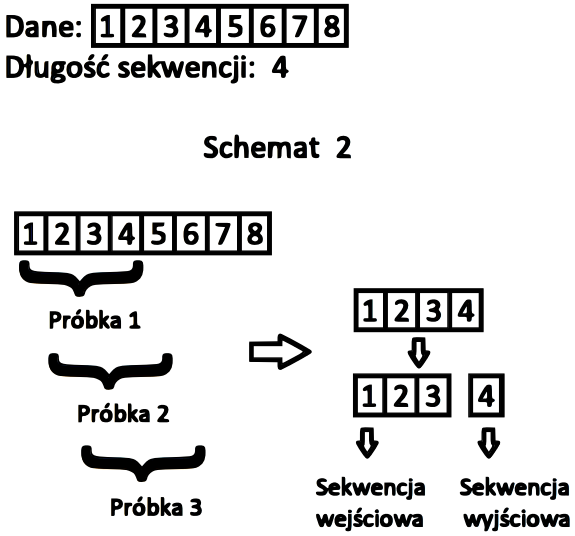
Następnym etapem przetwarzania zbioru uczącego jest kodowanie. Najpierw tworzony jest zbiór wszystkich nut i akordów, które przynajmniej raz wystąpiły w pozyskanym zbiorze. Każdej nucie i akordowi przypisywana jest unikalna liczba całkowita, która stanowi swego rodzaju identyfikator. Jest to ważny punkt, ponieważ zbiór danych pozyskany chwilę po przetworzeniu bazy treningowej, posiada jedynie elementy traktowane jako łańcuch znaków. Kodowanie pozwala na reprezentację tych danych jako liczby całkowite. Sieci neuronowe znacznie lepiej radzą sobie operując na danych numerycznych. Przekształcenie informacji tekstowej na reprezentację liczbową jest jednym z kluczowych kroków w kontekście uczenia maszynowego. Pozwala to modelowi na dużo bardziej efektywne przetwarzanie i analizę danych.

Po kodowaniu następuje podział danych na zbiór sekwencji wejściowych i wyjściowych. Stanowią one bazę do nauki modelu, odgrywają istotną rolę w procesie uczenia, ponieważ dostarczają modelowi informacji na temat oczekiwanych wyjść dla konkretnego wektora wejściowego. Sekwencje wejściowe to ciąg nut i akordów, które przekazywane są do modelu. Następnie podawana jest sekwencja wyjściowa, która w przypadku tego projektu stanowi pojedynczą nutę lub akord. Istnieją dwa główne sposoby generowania sekwencji. Pierwszy sposób polega na podziale danych poprzez przemieszczanie się o wartość długości sekwencji, co zostało pokazane na rysunku 3.1. Drugie podejście zakłada przesuwanie się o próbkę. Tego typu schemat powstawania sekwencji ukazany jest na rysunku 3.2. W tym projekcie zdecydowano się na implementację drugiego podejścia, ponieważ pozwa-

la ono wygenerować znacznie większą liczbę sekwencji, co jest ważne podczas trenowania modelu.



Rysunek 3.1 Tworzenie sekwencji poprzez przemieszczanie się o wartość długości sekwencji



Rysunek 3.2 Tworzenie sekwencji poprzez przesuwanie się o próbkę

Poprzez trening na tak przygotowanych danych, model uczy się różnorodnych wzorców muzycznych, które występują w danych wejściowych i wyjściowych. Taki sposób treningu umożliwia nauczenie się struktury melodii oraz wyłonienie pewnych zależności pomiędzy kolejnymi dźwiękami. Długość sekwencji wejściowej jest bardzo ważnym elementem, który należy dobrać w sposób empiryczny. Krótsze sekwencje wymagają mniej skomplikowanego modelu, co z kolei prowadzi do znacznie zmniejszonego zapotrzebowania na zasoby obliczeniowe oraz krótszy czas treningu sieci. Z drugiej strony dłuższa sekwencja ma istotny wpływ w uczeniu modelu bardziej złożonych wzorców w muzyce, co może być kluczowe w generowaniu bogatych i złożonych kompozycji. Im dłuższe sekwencje, tym dłużej trwa proces nauki oraz zwiększa się zapotrzebowanie na zasoby obliczeniowe.

Dalej następuje normalizacja czyli proces, w którym wszystkie wartości w wektorach wejściowych sprowadzane są do zakresu od 0 do 1. Dzięki normalizacji polepsza się proces

uczenia modelu. Trening staje się bardziej stabilny oraz skuteczniejszy. Dalej następuje przekształcanie danych wyjściowych za pomocą kodowania pozycyjnego (z ang. *One-Hot Encoding*). Polega to na reprezentowaniu każdej kategorii (akordu lub nuty) jako unikalnego wektora binarnego. Długość wektora jest taka sama jak liczba wszystkich unikalnych nut i akordów występujących w zbiorze danych. Każdy wektor składa się tylko i wyłącznie z jednej jedynki, natomiast pozostałe pola to zera. Tego typu reprezentacja ułatwia modelowi predykcję, ponieważ każda kategoria jest teraz reprezentowana jako jednoznaczny wektor binarny, co jest bardziej zrozumiałe dla sieci neuronowych.

Ostatnim etapem w przygotowaniu danych jest podzielenie zarówno sekwencji wejściowych jak i wyjściowych na dane treningowe oraz walidacyjne. Dane treningowe wykorzystywane są bezpośrednio do treningu modelu. Podczas nauki model korzysta z tych danych w celu aktualizacji wag w neuronach. Zbiór walidacyjny stanowi pewien wyznacznik tego, jak przebiega proces nauki sieci. Na jego podstawie możemy określić czy trening przebiega w sposób prawidłowy.

Rysunek 3.3 Ogólny schemat przetwarzania danych

3.3 Architektura sieci

Zaprojektowana rekurencyjna sieć neuronowa składa się z kilku warstw. Pierwszą z nich jest warstwa wejściowa (z ang. *input layer*). Składa się ona w całości z komórek rekurencyjnych. Jej zadaniem jest przyjęcie nowych danych wejściowych w postaci nut i akordów, następnie odpowiednie przetworzenie ich oraz przesłanie do kolejnych warstw. Wymaga

ona podania kilku kluczowych parametrów, które determinują jej funkcjonalność i skuteczność w kontekście poprawnego przetwarzania sekwencji dźwięków. Pierwszym z nich jest ilość sztucznych neuronów w warstwie. Odpowiedni dobór liczby komórek ma znaczny wpływ na zdolność modelu do nauki. Im większa ilość lub złożoność danych treningowych, tym więcej neuronów jest potrzebnych w warstwie wejściowej. Drugi parametr określa kształt danych wejściowych, które będą przesłane do modelu. W przypadku sieci rekurencyjnych, dane wejściowe są zazwyczaj reprezentowane przez trójwymiarowy wektor, w którym pierwszy wymiar określa liczbę sekwencji wejściowych, natomiast ich długość odpowiada wartości drugiego wymiaru. Trzeci wymiar reprezentuje liczbę cech.

Kolejne warstwy w sieci również składają się tylko z komórek rekurencyjnych. Stanowią one rdzeń modelu i odpowiedzialne są za wyodrębnienie z danych wejściowych istotnych cech sekwencji muzycznych. Tego typu podejście pozwala zyskać modelowi możliwość do uwzględnienia zależności czasowych, które występują w próbkach uczących. Dzięki temu sieć neuronowa może lepiej zrozumieć strukturę muzyczną oraz jej ewolucje w czasie. Podobnie jak w warstwie wejściowej, tutaj również ważne jest określenie dobrej liczby neuronów w komórce. Jednym ze sposobów na dobranie ilości komórek w warstwie jest ich stopniowe zwiększanie w kolejnych warstwach sieci. Pomaga to modelowi w hierarchicznym uczeniu się bardziej skomplikowanych cech na coraz wyższych poziomach abstrakcji. Następnie należy określić czy warstwa ma zwracać sekwencję dla każdego kroku czasowego czy tylko wartość ostatniej sekwencji. Dobrą praktyką jest zwracanie pełnych sekwencji w warstwie poprzedzającej warstwę rekurencyjną, ponieważ pozwala to na uwzględnienie pełnej informacji czasowej w kolejnych warstwach w modelu. Umożliwia to bardziej skuteczne przetwarzanie danych oraz lepsze zrozumienie i modelowanie zależności czasowych na różnych etapach przetwarzania. W przypadku modeli rekurencyjnych pełne sekwencje są często wymagane, aby skutecznie przenosić informacje pomiędzy kolejnymi krokami czasowymi. W warstwach złożonych z samych jednostek rekurencyjnych można uwzględnić dodatkowo opadanie rekurencyjne (z ang. *reccurent dropout*). Jest to jedna z wielu technik regularyzacji. Polega ona na losowym wyłączaniu wag w warstwach rekurencyjnych podczas treningu. W praktyce oznacza to, że w każdym kroku czasowym losowe połączenia pomiędzy neuronami są na pewien czas dezaktywowane. Celem tego typu technik jest zapobieganie przeuczeniu modelu. Zmuszanie sieci neuronowej do radzenia sobie z danymi bez nadmiernego wykorzystywania konkretnych połączeń może dodatkowo poprawić zdolność do generalizacji na nowe dane. Ważnym parametrem, który należy określić jest funkcja aktywacji. Jest ona przede wszystkim odpowiedzialna za wprowadzenie nieliniowości do sieci. Umożliwia to modelowi lepsze dopasowanie się do różnorodnych wzorców w danych treningowych. W praktyce oznacza to, że model jest w stanie rozwiązywać bardziej skomplikowane problemy. Popularną funkcją w warstwach składających się z komórek rekurencyjnych jest tangens hiperboliczny

$$\tanh(x) = \frac{e^{2x} - 1}{e^{2x} + 1}, \tag{3.1}$$

gdzie x jest wartością wejściową. Funkcja ta posiada dwie zasadnicze właściwości. Pierwsza z nich dotyczy zwracanych wartości w przedziale od (-1; 1). Jest to istotne w kontekście kontroli bramek w neuronach rekurencyjnych, takich jak LSTM czy GRU. Tangens hiperboliczny pozwala na efektywne sterowanie bramkami, dzięki czemu możliwe jest płynna regulacja przepływu informacji pomiędzy kolejnymi neuronami. Tanh jest symetryczny wokół zera, co pomaga w uwzględnieniu pozytywnych, jak i negatywnych (ujemnych) danych.

Po warstwach składających się z samych neuronów rekurencyjnych, występuje warstwa gęsta (z ang. *dense layer*). Cechą charakterystyczną tej warstwy jest fakt, iż wszystkie jej neurony są połączone z każdym neuronem z poprzedniej, jak i następnej warstwy. To oznacza, że każdy neuron w warstwie gęstej otrzymuje na swoje wejście wszystkie wyjścia neuronów z poprzedniej warstwy i przesyła swoje wyjście do wszystkich neuronów w następnej warstwie. Podobnie jak w poprzednich przypadkach, ważnym krokiem jest dobór parametrów warstwy. Pierwszy parametr odnosi się do liczby neuronów. Zbyt mała ilość komórek może sprawić, że model nie będzie w stanie nauczyć się złożonych zależności. Zazwyczaj dobór tego parametru odbywa się poprzez eksperymenty, polegające na stopniowym zwiększaniu ilości neuronów i monitorowaniu jakości modelu np. na zbiorze walidacyjnym. Drugi parametr odnosi się do funkcji aktywacji. Jednym z najpopularniejszych wyborów jest rektyfikowana jednostka liniowa (z ang. *rectified linear unit*)

$$ReLU(x) = \max(0, x), \tag{3.2}$$

gdzie x oznacza wartość wejściową. Oprócz wprowadzania nieliniowości funkcja ReLU stanowi rozwiązanie problemu zanikającego gradientu. Zapobiega ona wykładniczemu wzrostowi w obliczeniach, które są wykorzystywane podczas działania sieci. Zastosowanie tego typu funkcji praktycznie uniemożliwia zbliżenie się gradientu komórki do zera.

Ostatnią warstwą jest warstwa wyjściowa (z ang. *output layer*), która w tym przypadku jest również warstwą gęstą. Liczba neuronów w warstwie wyjściowej dla tego typu projektu jest równa liczbie klas (unikalnych nut i akordów) występujących w danym zbiorze treningowym. Oznacza to, że model może generować jedynie dźwięki, które przynajmniej raz pojawiły się na etapie treningu. Jako funkcję aktywacji wybrano operację miękkiego maksimum (z ang. *soft max*)

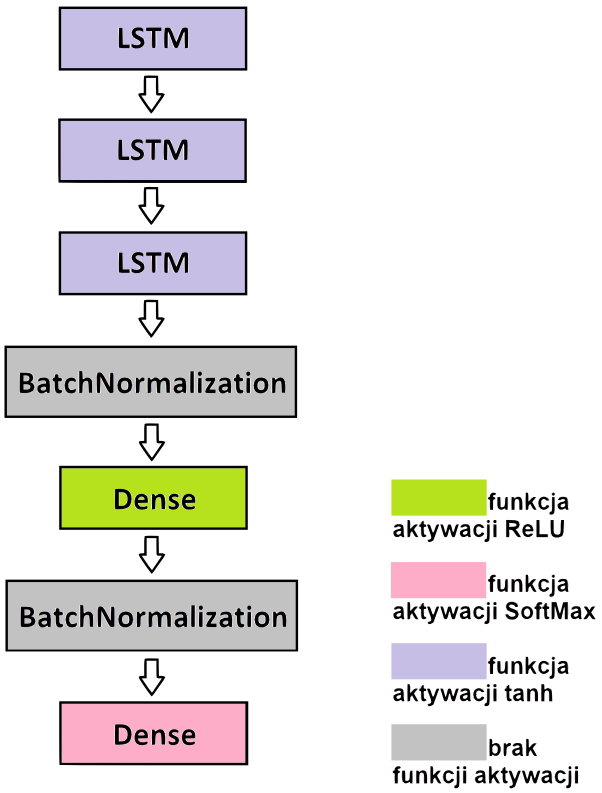
$$x_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}, \tag{3.3}$$

gdzie x_i określa prawdopodobieństwo przypisane i-temu elementowi wektora z po zastosowaniu funkcji Softmax, $\sum_{j=1}^{K} e^{z_j}$ jest sumą eksponent dla wszystkich elementów wektora z, natomiast e^{z_i} to funkcja eksponenty dla i-tego elementu wektora z. Zadaniem funkcji Softmax jest przekształcenie wyników z warstwy wyjściowej modelu na prawdopodobieństwo przynależności do różnych klas. Każda klasa otrzymuje swoje prawdopodobieństwo, natomiast suma wszystkich prawdopodobieństw wynosi 1.

Podczas nauki sieci neuronowej może dojść do przetrenowania (z ang. *overfitting*). Zjawisko to polega na wyuczeniu się zbioru danych treningowych. Jedną z przyczyn overfittingu może być zbyt duża ilość epok, ponieważ długotrwałe trenowanie modelu na tych samych danych może skutkować dostosowaniem się do przypadkowych detali, które nie mają istotnego znaczenia. Istnieje kilka metod, które pomagają wyeliminować to zjawisko. Pierwsza z nich – opadanie rekurencyjne, została już wcześniej poruszona. Inną metodą może być stosowanie warstwy porzucenia (z ang. *dropout layer*). Jej sposób działania jest bardzo podobny jak w przypadku opadania rekurencyjnego z tą różnicą, że opadanie rekurencyjne wyłącza losowo połączenia między neuronami, natomiast warstwa porzucenia dezaktywuje w sposób losowy część neuronów. Kolejną metodą jest normalizacja Batchowa (z ang. *Batch normalization*). W trakcie treningu normalizowane są wartości wejściowe do warstw poprzez średnią i odchylenie standardowe dla danego batcha. Następnie dane są przeskalowywane i przesuwane, przy użyciu dwóch parametrów γ i β, które są aktualizowane w trakcie treningu. Skalowanie pozwala na przywrócenie elastyczności modelu. Dodatkową zaletą tej techniki jest przyspieszenie i stabilizacja procesu uczenia modelu.

3.4 Zaimplementowane modele

W ramach projektu zostały zaimplementowane dwie sieci neuronowe. Pierwszy model wykorzystuje komórki LSTM, jako wcześniej omawiane neurony rekurencyjne. Ogólny schemat budowy wyżej wymienionego modelu znajduje się na rysunku 3.4. Odczytać można z niego kolejność poszczególnych warstw, co również mówi o kierunku przepływu informacji w sieci.



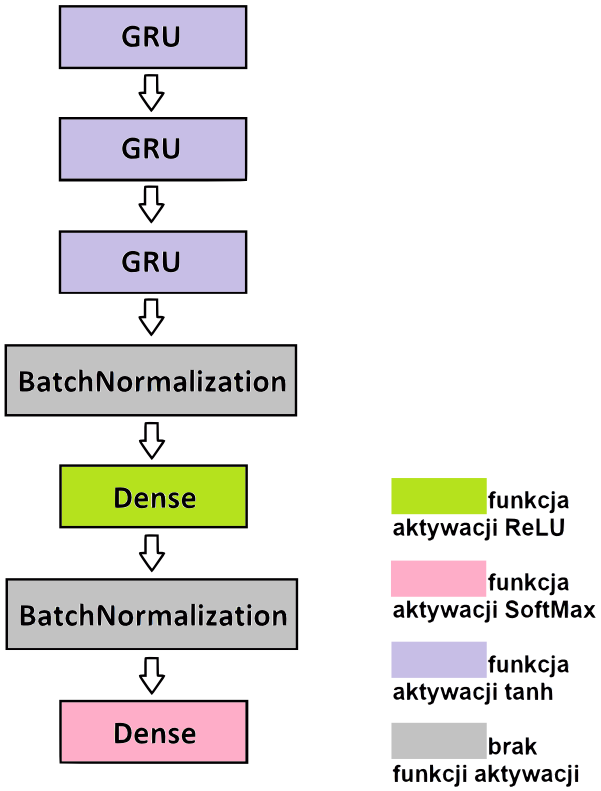
Rysunek 3.4 Schemat zaimplementowanej architektury z komórkami LSTM

Dodatkowo w tabeli 3.1 umieszczone są bardziej szczegółowe informacje dotyczące budowy konkretnych warstw takie jak np. ilość neuronów.

Tabela 3.1 Parametry warstw sieci z komórkami LSTM

Warstwa	Liczba komórek	Sekwencje	Porzucenie	Opadanie rekurencyjne
LSTM	256	Tak	-	0.3
LSTM	512	Tak	-	0.3
LSTM	512	Nie	-	-
BatchNormalization	-	-	0.3	-
Dense	256	-	-	-
BatchNormalization	-	-	0.3	
Dense	Liczba unikalnych dźwięków	-	-	-

Druga sieć neuronowa wykorzystuje komórki GRU. Analogiczny schemat budowy drugiego modelu znajduje się na rysunku 3.5, natomiast bardziej szczegółowe informacje dotyczące poszczególnych warstw ukazane są w tabeli 3.2.



Rysunek 3.5 Schemat zaimplementowanej architektury z komórkami GRU

Tabela 3.2 Parametry warstw modelu z komórkami GRU

Warstwa	Liczba komórek	Sekwencje	Porzucenie	Opadanie rekurencyjne
GRU	256	Tak	-	0.3
GRU	512	Tak	-	0.3
GRU	512	Nie	-	-
BatchNormalization	-	-	0.3	-
Dense	256	-	-	-
BatchNormalization	-	-	0.3	
Dense	Liczba unikalnych dźwięków	-	-	-

Jak można zauważyć zaimplementowane modele różnią się jedynie rodzajem wykorzystywanych neuronów rekurencyjnych. Tego typu podejście pozwala porównać wpływ poszczególnych komórek na jakość generowanych melodii country.

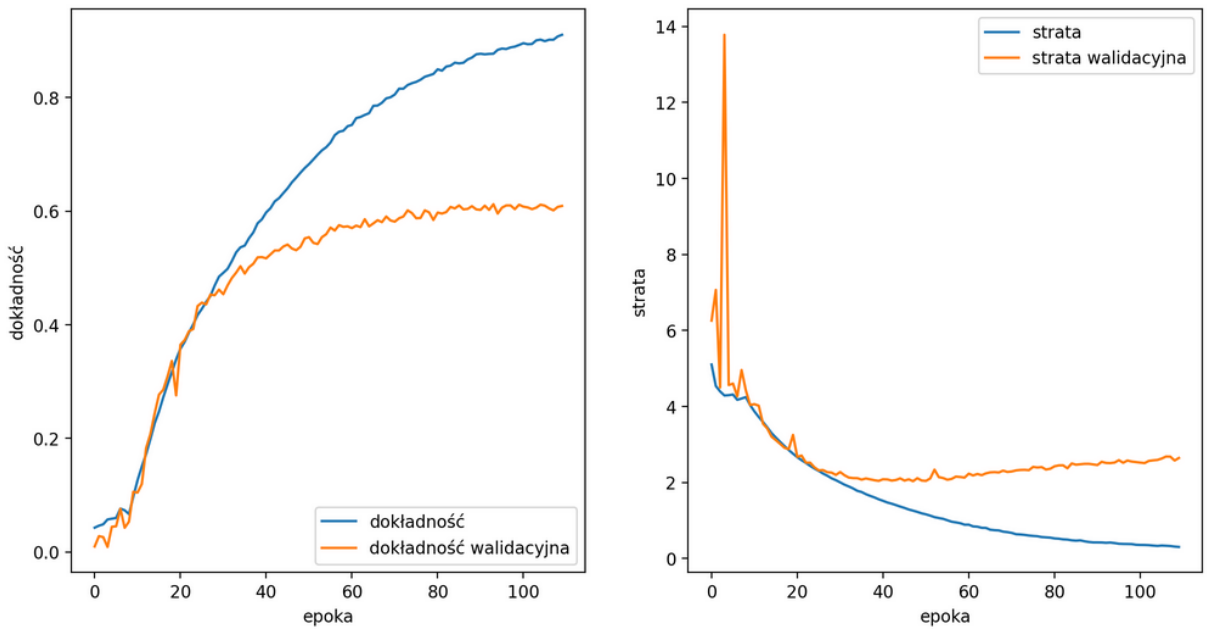
3.5 Uczenie sieci

Celem uczenia sieci neuronowej jest dostosowywanie wag neuronów, tak aby model był w stanie jak najlepiej realizować postawione przed nim zadanie. W praktyce polega to na dobraniu takich wag połączeń pomiędzy komórkami, aby jak najbardziej zminimalizować wartość funkcji straty (z ang. *loss function*). Funkcja ta określa stopień w jakim prognozy modelu różnią się od rzeczywistych wartości. Wybór odpowiedniej funkcji straty do postawionego zadania jest ważnym czynnikiem, który ma duży wpływ na jakość predykcji sieci. W przypadku tego projektu zdecydowano się na wybór kategorycznej entropii krzyżowej (z ang. *categorical crossentropy*), ponieważ jest to funkcja, która dobrze nadaje się do problemów związanych z klasyfikacją, gdzie model ma za zadanie przyporządkować

uzyskane dane wyjściowe do jednej z wielu klas. Kategoryczna entropia krzyżowa mierzy, jak bardzo rozkład prawdopodobieństwa przewidywany przez model różni się od rzeczywistego rozkładu klas w danych treningowych czy też walidacyjnych. Minimalizacja tej funkcji straty sprawia, że prognozy modelu zbliżają się do oczekiwanych wyjść.

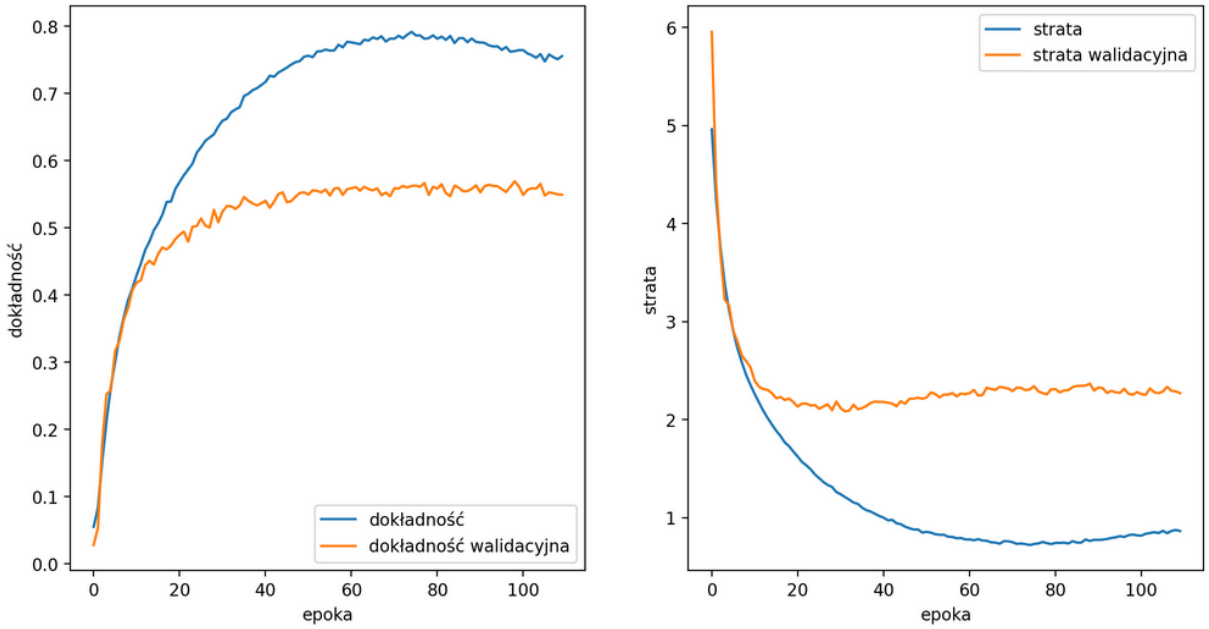
Kolejnym kluczowym aspektem do przeprowadzenia poprawnego uczenia sieci jest dobranie optymalizatora. Jego głównym zadaniem jest minimalizacja funkcji straty. Optymalizatory są odpowiedzialne za skuteczne dostosowywanie parametrów modelu, szczególnie wag połączeń między neuronami, aby model był w stanie dokonywać bardziej trafnych predykcji. Opierają się one na zasadzie znajdowania minimum funkcji straty. W praktyce oznacza to znalezienie konfiguracji wag, dla których różnica pomiędzy prognozami modelu, a rzeczywistymi wartościami jest minimalna. W tym projekcie zdecydowano się na optymalizator propagacji średniej kwadratowej (z ang. *root mean square propagation, RMSprop*), który jest jednym z najpopularniejszych optymalizatorów w kontekście rekurencyjnych sieci neuronowych. RMSprop jest adaptacyjnym optymalizatorem, który dobrze radzi sobie z problemem dostosowywania szybkości uczenia do różnych parametrów w sieci. Działa poprzez uwzględnienie historii kwadratów gradientów, dzięki czemu jest on w stanie bardziej elastycznie aktualizować wagi w sieci.

W przypadku tego projektu oba zaimplementowane modele przechodziły proces nauki wykorzystując tę samą funkcję straty (kategoryczna entropia krzyżowa) jak i ten sam optymalizator (RMSprop). Miało to na celu lepsze porównanie uzyskanych wyników. Po zakończonym treningu przystąpiono do oceny procesu uczenia modeli. W tym celu zostały wygenerowane wykresy przedstawiające przebieg funkcji dokładności oraz straty. Na rysunku 3.6 zobaczyć można przebieg wartości funkcji straty dla sieci wykorzystującej komórki LSTM. Przebieg oznaczony jako funkcja strat dla walidacji określa wartość funkcji straty dla zbioru walidacyjnego, natomiast strata odnosi się do próbek treningowych. Na tym samym rysunku przedstawiono również wykres dokładności. Przebieg określony jako dokładność odnosi się do zbioru treningowego, dokładność walidacyjna dotyczy danych walidacyjnych. Wykresy dokładności przedstawiają stosunek przypadków dobrze sklasyfikowanych przez model do liczby wszystkich klasyfikacji podczas nauki.



Rysunek 3.6 Analiza procesu ucznia modelu złożonego z komórek LSTM

Analogiczne wykresy zostały stworzone dla modelu zawierającego komórki GRU. Na rysunku 3.7 znajdują się wykresy wartości funkcji straty oraz dokładności.



Rysunek 3.7 Analiza procesu ucznia modelu złożonego z komórek GRU

Z wykresów możemy odczytać, iż model z komórkami LSTM osiągnął większą dokładność w porównaniu z drugą siecią. Dodatkowo sieć wykorzystująca neurony LSTM osiągnęła mniejszą wartość funkcji straty. W przypadku funkcji straty walidacyjnej oraz dokładności walidacyjnej, oba modele osiągnęły bardzo zbliżone wyniki. Analizując rysunki można stwierdzić, że proces nauki zarówno dla pierwszego, jak i drugiego modelu przebiegł prawidłowo. Wraz z kolejnymi epokami wartości funkcji straty miały zazwyczaj tendencję malejącą, natomiast dokładność zaimplementowanych sieci miała charakter przeważnie wzrostowy. Obie charakterystyki zgodnie z oczekiwaniami ustabilizowały się mniej więcej na danym poziomie. Na żadnym z etapów treningu nie doszło do załamania się mierzonych charakterystyk.

Rozdział 4

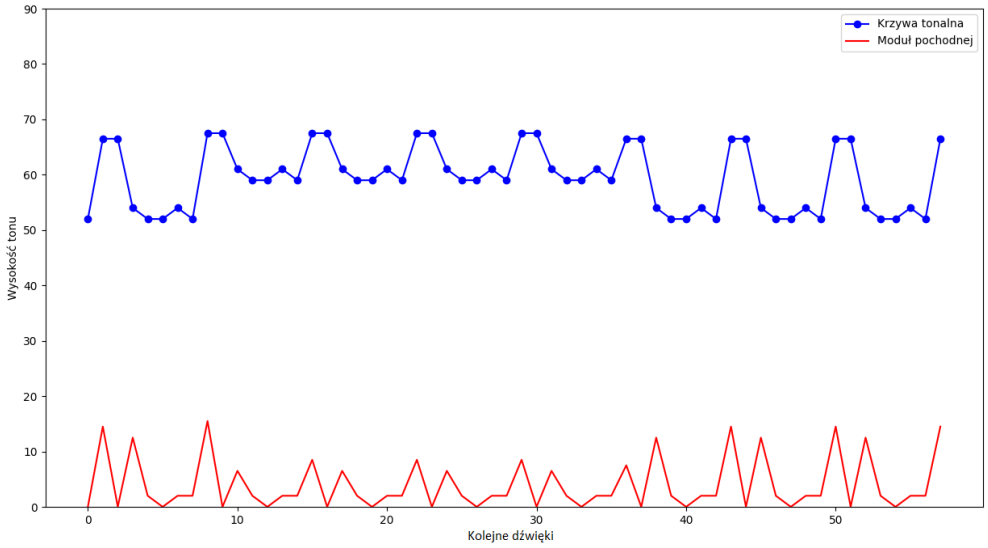
Analiza utworów

W celu oceny jakości melodii generowanych przez model zdecydowano się na użycie trzech technik, które pozwolą stwierdzić czy wytrenowana sieć jest w stanie poprawnie wykonywać postawione przed nią zadanie. W przypadku tego projektu standardowe techniki oceny jakości, wytrenowanego modelu takie jak analiza wartości funkcji straty czy precyzji nie pozwalają jednoznacznie określić czy sieć jest w stanie samodzielnie tworzyć melodię country. Powodem takiego stanu rzeczy jest fakt, iż w zadaniu generowania melodii określonego gatunku nie istnieje jedno prawidłowe wyjście, ale wiele możliwych dźwięków, które w danej chwili będą zgodne z duchem oczekiwanego stylu muzycznego. Pierwsza wykorzystana technika analizy utworów polega na wykreśleniu krzywej tonalnej. Pokazuje ona wartości tonów kolejnych dźwięków w melodii. Analizując krzywą tonalną możemy określić jaki zakres tonów jest charakterystyczny dla danego gatunku muzycznego. Drugim narzędziem jest moduł pochodnej krzywej tonalnej, która ilustruje charakter zmian tonów. Wykorzystując moduł pochodnej można określić tempo zmian tonów dla danej sekwencji dźwięków. Ostatnią wykorzystaną techniką oceny gatunku muzyki jest szybka transformacja Fouriera (z ang. *Fast Fourier Transform, FFT*). Poprzez przekształcenie sygnału z dziedziny czasu do dziedziny częstotliwości, FFT pozwala na uzyskanie spektralnej reprezentacji dźwięku. W rezultacie jesteśmy w stanie określić, które częstotliwości są charakterystyczne dla danego utworu.

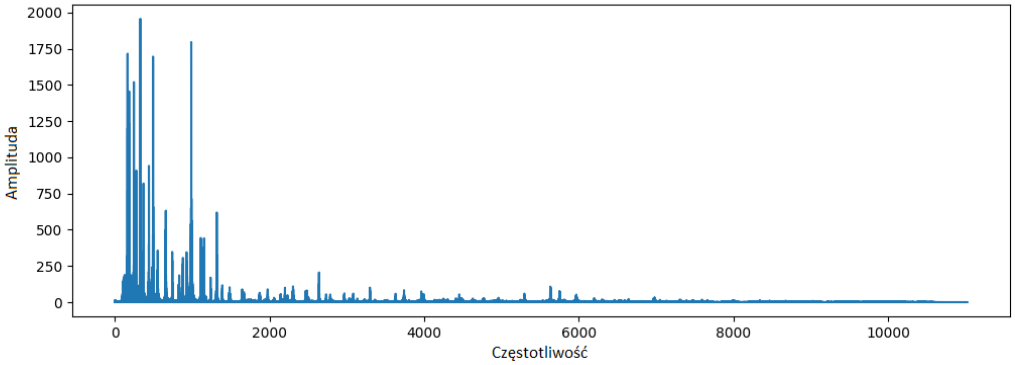
4.1 Oryginalne utwory country

Analiza oryginalnych utworów country stanowi punkt wyjścia w ocenie jakości melodii generowanych przez sieć. W tym celu poddano badaniom wiele autentycznych kompozycji muzycznych, które wchodziły w skład bazy uczącej. Poniżej przedstawiono wyniki analiz dla przykładowych dwóch utworów. Na rysunkach 4.1 i 4.2 ukazane są kolejno wykres krzywej tonalnej wraz z modułem jej pochodnej oraz analiza FFT dla fragmentu pierwszej melodii. Analogiczne wykresy dla części drugiego oryginalnego utworu zobaczyć można na rysunkach 4.3 i 4.4. Tego typu analiza pozwoliła na określenie charakterystycznych cech melodii country. Ustalono następujące właściwości badanych utworów, które jednocześnie uznano za kryteria jakie powinna spełniać generowana melodia country:

- przebieg krzywej tonalnej powinien w całości zawierać się w przedziale od wysokości tonu równego 50 do poziomu 73,
- moduł pochodnej krzywej tonalnej nie powinien przekroczyć wartości 17,
- w utworach powinny dominować częstotliwości z zakresu od 80Hz do 1800Hz.



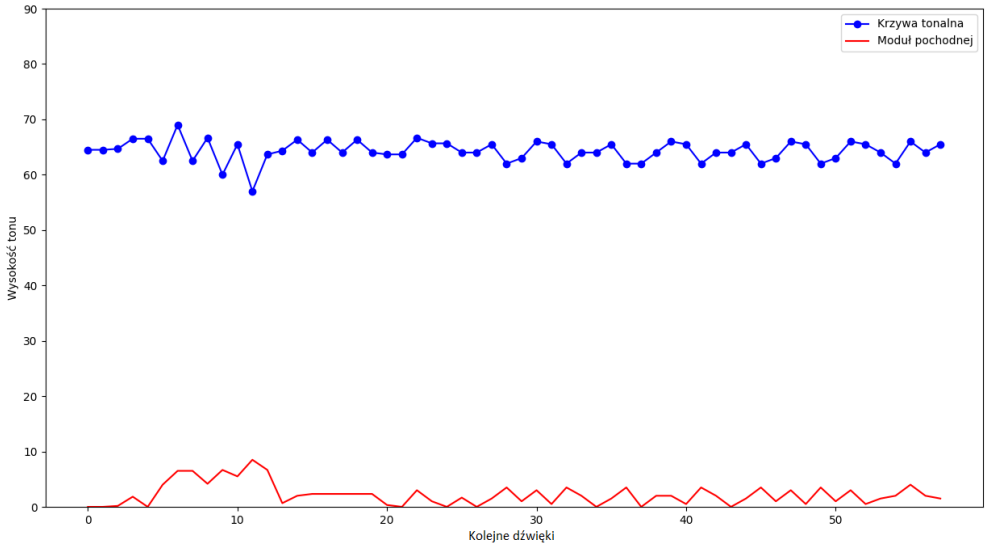
Rysunek 4.1 Krzywa tonalna wraz z modułem jej pochodnej wykreślona dla fragmentu oryginalnej melodii



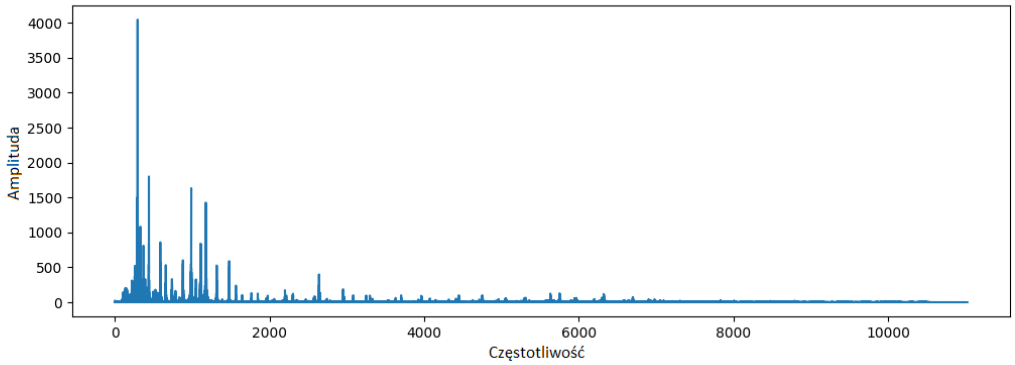
Rysunek 4.2 Analiza FFT przeprowadzona na fragmencie oryginalnej melodii

4.2 Utwory wygenerowane przez modele

Taką samą analizę przeprowadzono dla kompozycji wygenerowanych przez wytrenowane modele. Ważnym elementem podczas generowania utworów jest wybór sekwencji wejściowej. Jej charakter ma wpływ na czas potrzebny do ustabilizowania się modelu. W praktyce oznacza to jak długo musimy czekać, aż sieć zacznie generować pożądane melodie. Standardowe podejście polega na użyciu danych testowych jako sekwencji wejściowej. Są to dane, z którymi model nigdy wcześniej nie miał do czynienia. Dodatkowo mają one taki sam charakter jak próbki, na których przeprowadzony był proces nauki modelu. Dzięki temu model nie potrzebuje czasu na ustabilizowanie się i zaczyna od razu generować pożądane kompozycje muzyczne. Drugi sposób polega na podaniu losowych wartości jako sekwencji wejściowych. Tego typu podejście pozwala na lepsze sprawdzenie tego czy model faktycznie nauczył się pożądanych cech kompozycji muzycznej i jest w stanie samodzielnie generować utwory danego gatunku. Wadą tej metody jest czas, który model potrzebuje do ustabilizowania wyjścia i rozpoczęcia generowania oczekiwanych utworów. W pierwszym



Rysunek 4.3 Krzywa tonalna wraz z modułem jej pochodnej wykreślona dla fragmentu drugiej oryginalnej melodii



Rysunek 4.4 Analiza FFT przeprowadzona na fragmencie drugiej oryginalnej melodii

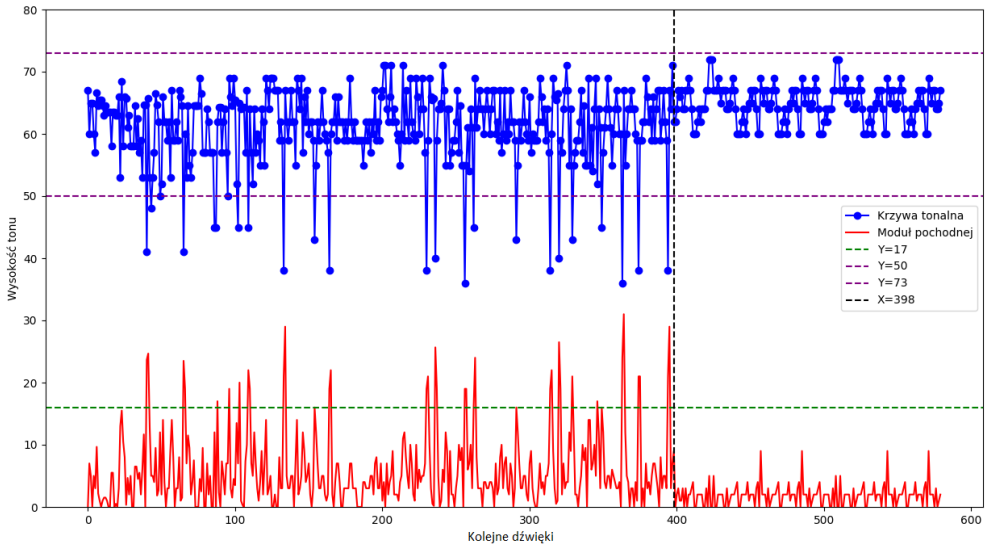
podejściu istnieje możliwość, że sieć będzie jedynie naśladować cechy utwory wejściowego. W przypadku tego projektu zdecydowano się na zastosowanie drugiego podejścia. Jako sekwencje wejściowe modelu użyto białego szumu. Jego dużą zaletą jest fakt, że wszystkie częstotliwości występujące w danym zakresie mają taką samą moc.

Dla obu modeli zostały wygenerowane oraz przebadane utwory. W dalszej części pracy można zobaczyć wyniki analizy dla dwóch utworów stworzonych przez model złożony z komórek LSTM. Na rysunkach 4.5 i 4.6 przedstawiony jest wykres krzywej tonalnej wraz z modułem jej pochodnej i analiza FFT przeprowadzona dla fragmentu pierwszego utworu. Dalej na rysunkach 4.7 i 4.8 przedstawione są wyniki analizy dla drugiej melodii. Dokładnie takie same badania dla modelu złożonego z komórek GRU zobaczyć można na rysunkach 4.9 i 4.10 dla pierwszego utworu, natomiast analiza drugiej kompozycji widoczna jest rysunkach 4.11 i 4.12. Jak można zauważyć na poniższych wykresach zostały dodane linie, które pomagają ocenić dane przebiegi. Przerywana zielona linia oznacza granicę, za którą nie powinien znaleźć się moduł pochodnej krzywej tonalnej, natomiast

sam pożądany zakres występowania krzywej tonalnej został ograniczony przerywanymi fioletowymi liniami.

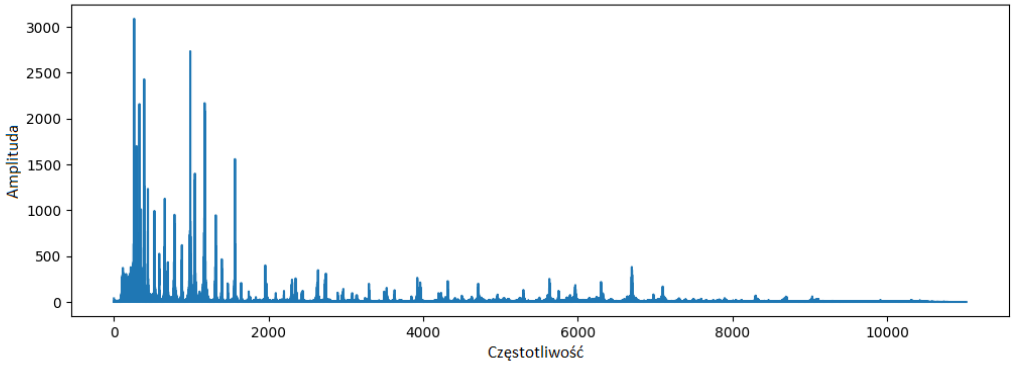
W pierwszym kroku skupiono się na zlokalizowaniu momentu, od którego utwór zaczyna spełniać wymogi związane z krzywą tonalną i modułem jej pochodnej. Moment ten zaznaczony jest pionową czarną przerywaną linią. Oznacza to, że fragment utworu położony na lewo od czarnej przerywanej linii nie jest oczekiwanym rezultatem i nie był w dalszej części badań brany pod uwagę. Dźwięki położone na prawo spełniają założone kryteria dotyczące krzywej tonalnej. Następnie część ta (położona na prawo od czarnej przerywanej linii) poddawana była analizie FFT, aby zweryfikować ostatnie wymaganie dotyczące oczekiwanej jakości generowanych melodii. Jeśli w wyniku wyżej wymienionej analizy okazało się, że w podanym fragmencie utworu dominują częstotliwości z zakresu od 80Hz do 1800Hz to oznacza, że model poprawnie realizuje zadanie związane z generowaniem melodii country.

Analiza wykresów krzywych tonalnych związanych z modelami wykorzystującymi komórki LSTM pozwala dostrzec kilka wzorców. Po pierwsze model w pewnym momencie dochodzi do stanu, w którym zaczyna generować oczekiwane przebiegi obu krzywych (rysunki 4.5, 4.7). Analiza FFT przeprowadzona na melodiach, których dotyczą wyżej wymienione przebiegi (wykresy 4.6 i 4.8) pokazuje, że w utworach tych dominują częstotliwości z zakresu od 80Hz do 1800Hz. Stanowi to dowód tego, iż model potrafi samodzielnie gene-



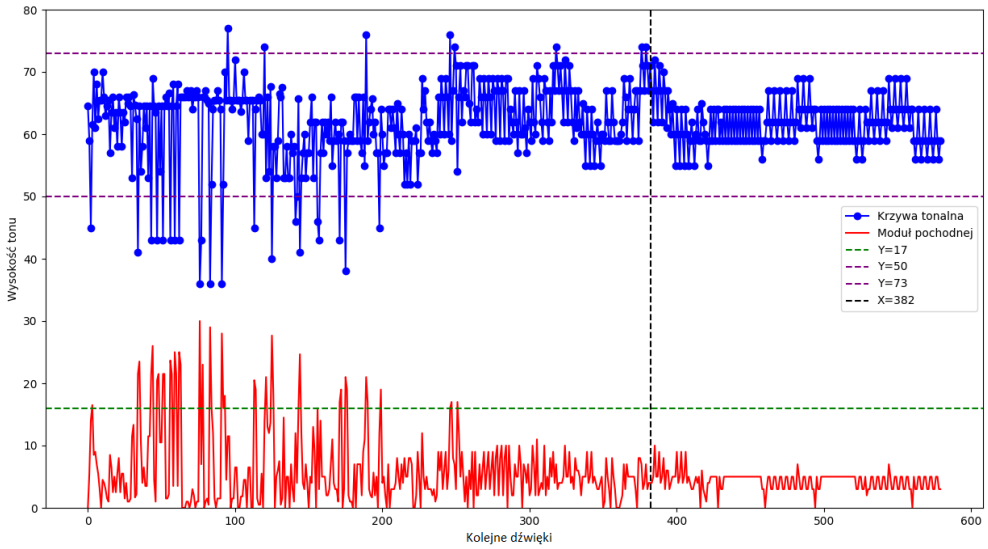
Rysunek 4.5 Krzywa tonalna wraz z modułem jej pochodnej wykreślona dla pierwszego utworu wygenerowanego przez model z komórkami LSTM

rować autentyczne melodie country, a nie jedynie naśladować charakter sekwencji wejściowych. Dodatkowo można dostrzec, że sieć musi wygenerować około 400 nut lub akordów, zanim w utworze wyklarują się pożądane cechy muzyki country. Innym ciekawym faktem jest to, że model ma tendencję głównie do przekraczania dolnej granicy dla krzywej tonalnej. Co prawda na rysunku 4.7, można również zobaczyć kilkukrotne przekroczenie górnej granicy przez krzywą tonalną, natomiast analiza większej ilości wygenerowanych utworów pokazała, że jest to zjawisko występujące znacznie rzadziej. Przypatrując się wartościom funkcji tonalnej można zauważyć, iż znaczna większość dźwięków generowanych przez model spełnia zdefiniowane wcześniej kryteria dotyczące wartości tonów, jednak raz na jakiś



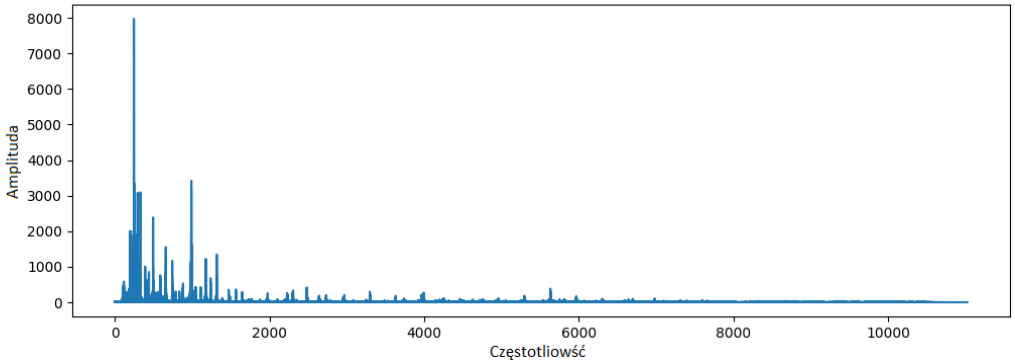
Rysunek 4.6 Analiza FFT przeprowadzona dla fragmentu pierwszego utworu wygenerowanego przez model z komórkami LSTM

czas powstają swego rodzaju anomalie czyli dźwięki, których poziom tonu znacznie odbiega od oczekiwanej wysokości. Proces stabilizacji modelu polega właśnie na pozbyciu się tych anomalii. Analizując pochodną krzywej tonalnej można zauważyć, że do czasu



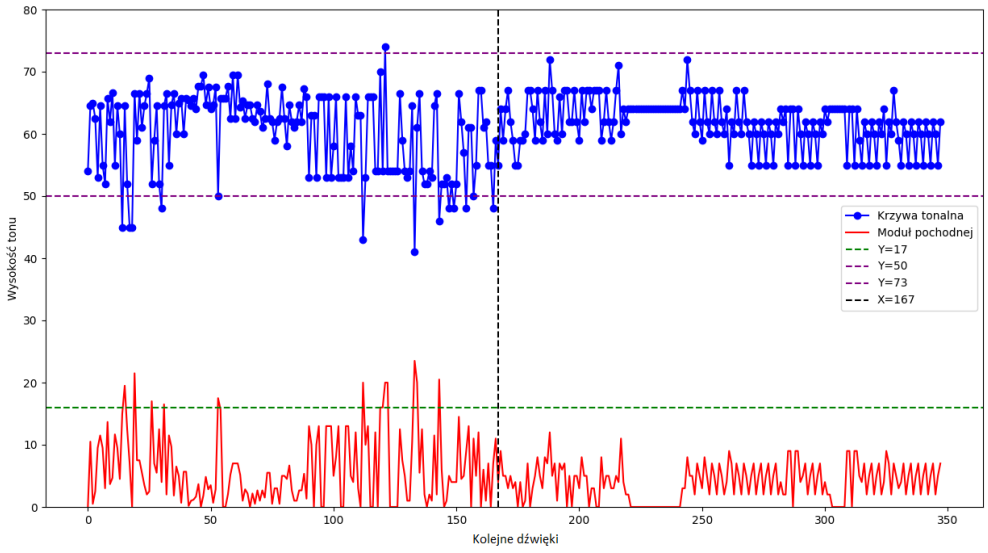
Rysunek 4.7 Krzywa tonalna wraz z modułem jej pochodnej wykreślona dla drugiego utworu wygenerowanego przez model z komórkami LSTM

wyklarowania się oczekiwanej melodii, model ma tendencję do tworzenia kolejnych dźwięków, których wysokość tonu jest od siebie skrajnie różna. Na rysunku 4.5 zobaczyć można, że w pewnym momencie pochodna krzywej tonalnej osiąga wartości większe i równe 30. Obserwując krzywą tonalną jak i jej pochodną znajdującą się na prawo od czarnej przerywanej linii, możemy dostrzec kilka motywów muzycznych, które regularnie się powtarzają. Świadczy to o jedności i spójności melodii.



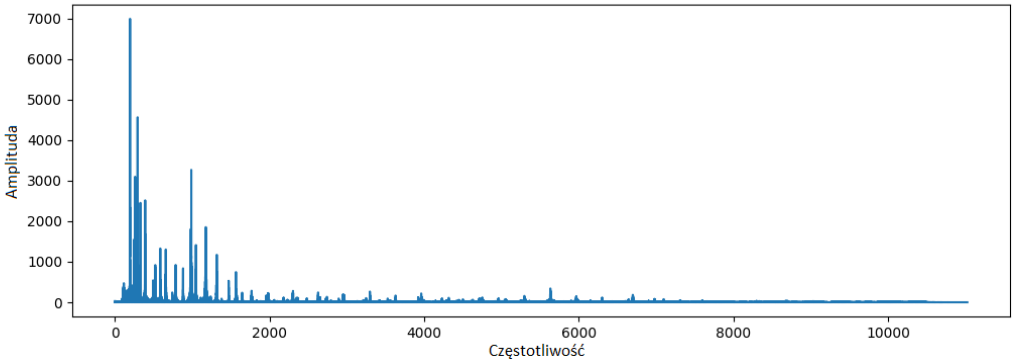
Rysunek 4.8 Analiza FFT przeprowadzona dla fragmentu drugiego utworu wygenerowanego przez model z komórkami LSTM

Badania utworów wygenerowanych przez sieć z komórkami GRU dostarczają podobne wnioski jak w poprzednim przypadku, chociaż głębsza analiza przebiegów pokazuje, że istnieje również kilka różnic. Najważniejsze jest to, że model dochodzi do stanu, w którym zaczyna tworzyć melodię spełniającą kryteria odnośnie krzywej tonalnej oraz jej pochodnej (rysunki 4.9 oraz 4.11). Dalsza analiza FFT przeprowadzona dla fragmentów, których



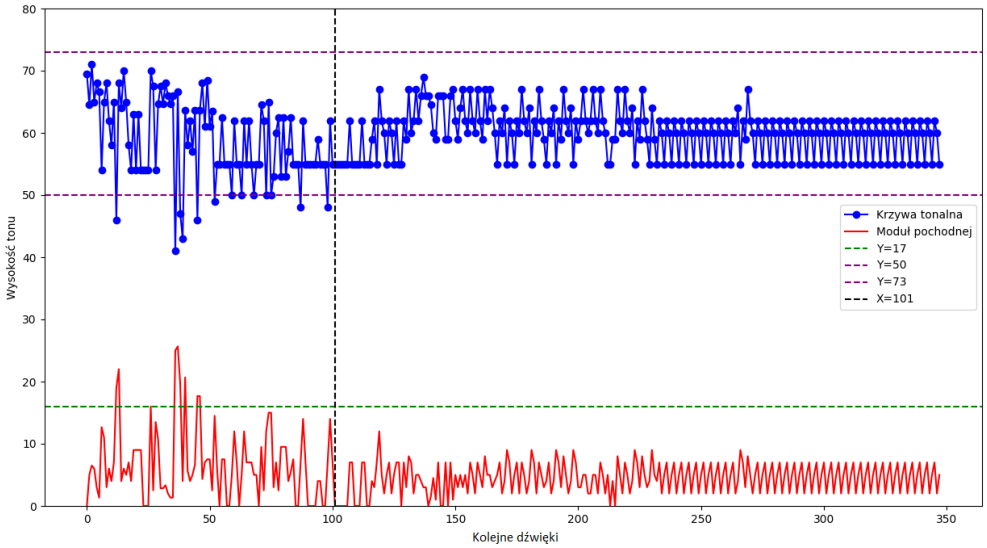
Rysunek 4.9 Krzywa tonalna wraz z modułem jej pochodnej wykreślona dla pierwszego utworu wygenerowanego przez model z komórkami GRU

krzywe tonalne spełniają postawione wymagania pokazuje, że w utworach tych dominują częstotliwości z zakresu od 80Hz do 1800Hz (rysunki 4.10 i 4.12). Zatem można stwierdzić, iż model zawierający komórki GRU jest w stanie samodzielnie tworzyć autentyczne utwory country. W przeciwieństwie do wcześniej omawianego modelu, sieć z komórkami GRU potrafi znacznie szybciej wyklarować pożądane cechy generowanych melodii. W przypadku poniższych przykładów model potrzebował 101 oraz 167 dźwięków do rozpoczęcia tworzenia oczekiwanych utworów. W porównaniu z modelem złożonym z komórek LSTM, który wymagał około 400 nut lub akordów jest to wyraźna różnica. W tym przypadku



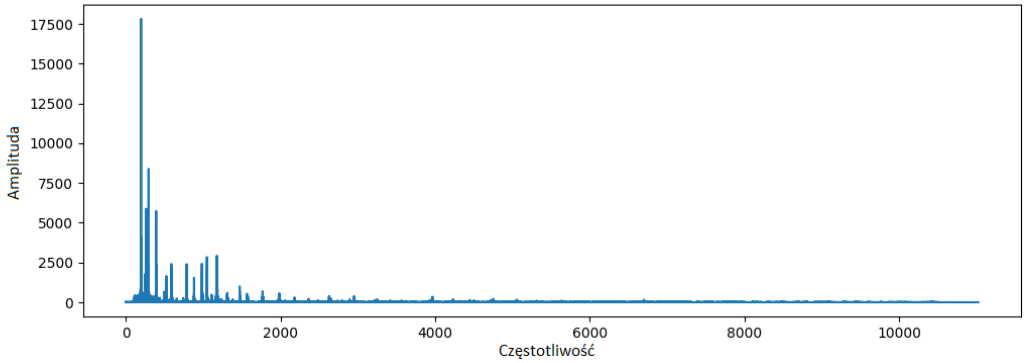
Rysunek 4.10 Analiza FFT przeprowadzona dla fragmentu pierwszego utworu wygenerowanego przez model z komórkami GRU

również, wytrenowana sieć ma wyraźne tendencje do przekraczania dolnej granicy krzywej tonalnej. Podobnie jak wcześniej znaczna większość generowanych dźwięków spełnia założone kryteria jednak co jakiś czas model generuje nuty i akordy, znacząco odbiegające od oczekiwanych dźwięków. Analiza pochodnej krzywej tonalnej pokazuje, że analizowany model nie ma tak dużej tendencji do tworzenia dźwięków, których wysokość tonu jest od siebie skrajnie różna. W wyniku analizy stwierdzono, że pochodna rzadko dochodzi do wartości 25, chociaż jak można zobaczyć na rysunku 4.11 zdarzają się takie przypadki. Model z komórkami GRU posiada tendencję do zapętlania. Przykład takiego zjawiska wi-



Rysunek 4.11 Krzywa tonalna wraz z modułem jej pochodnej wykreślona dla drugiego utworu wygenerowanego przez model z komórkami GRU

doczny jest na rysunku 4.9 gdzie widać, że między 200 a 250 dźwiękiem występuje szereg takich samych nut lub akordów (pozioma linia na wykresie krzywej tonalnej). Obserwując krzywą tonalną jak i jej pochodną znajdującą się na prawo od czarnej przerywanej linii możemy stwierdzić, iż tutaj trudniej jest dostrzec różne powtarzające się schematy. Zatem nie świadczy to dobrze o spójności utworu.



Rysunek 4.12 Analiza FFT przeprowadzona dla fragmentu drugiego utworu wygenerowanego przez model z komórkami GRU

Rozdział 5

Podsumowanie

Założenia dotyczące pracy inżynierskiej zostały zrealizowane. W rezultacie udało się stworzyć dwa modele sieci neuronowych, które na podstawie wejściowej sekwencji dźwiękowej są w stanie generować autentyczne melodie w stylu country. Projekt ten wymagał realizacji szeregu kamieni milowych. Pierwszy polegał na skompletowaniu zestawu odpowiednich danych, które stanowią bazę uczącą sieci. Dalej został opracowany proces przetwarzania wcześniej wymienionych danych, od wyodrębnienia konkretnej ścieżki instrumentu, przez kodowanie danych, aż do wyboru odpowiedniej metody podziału próbek uczących na sekwencje wejściowe i wyjściowe. Następny krok polegał na doborze architektury sieci. Było to najtrudniejsze zadanie, ponieważ wymagało ono znalezienia kompromisu pomiędzy ilością oraz złożonością danych wejściowych, wymaganymi zasobami obliczeniowymi, czasem potrzebnym na naukę modelu, a jakością generowanych utworów przez modele. Kolejnym krokiem był dobór odpowiednich parametrów dotyczących procesu nauki sieci takich jak optymalizator czy funkcja straty. Parametry te miały duży wpływ na ostateczny kształt wytrenowanych modeli, a co za tym idzie również na jakość generowanych utworów. Kolejny etap prac polegał na dobraniu oraz opracowaniu odpowiednich narzędzi, które pozwalają ocenić generowane melodie w sposób analityczny. Ostatnią czynnością była analiza i porównanie oryginalnych utworów country z melodiami wygenerowanymi przez modele.

W pracy udało się udowodnić, iż rodzaj wykorzystanych komórek rekurencyjnych ma wpływ na jakość tworzonych utworów. W porównaniu do sieci złożonej z komórek GRU, model wykorzystujący komórki LSTM potrzebuje znacznie więcej czasu, aby zacząć generować autentyczne melodie w stylu country. Analiza utworów, które spełniają kryteria dotyczące pożądanych cech country pokazuje, że melodie generowane przy wykorzystaniu komórek LSTM są spójniejsze oraz bardziej jednolite. Zaobserwować to można na przebiegach modułu pochodnej krzywej tonalnej, gdzie widać różne przeplatające się schematy muzyczne. W przypadku sieci z komórkami GRU trudniej jest zaobserwować tego typu zależności. Dodatkowo model ten posiada tendencje do zapętlania się co zdecydowanie nie jest pożądaną cechą, ponieważ obniża to różnorodność utworu. Przyczyną takiego stanu rzeczy może być fakt, iż neurony LSTM mają bardziej złożoną budowę, dzięki czemu mogą sprawniej przetwarzać skomplikowane zależności. Przypuszcza się, że aby uzyskać podobne rezultaty dla sieci złożonej z neuronów GRU należy zastosować bardziej pogłębioną architekturę tzn. zwiększyć liczbę warstw oraz ilość neuronów. Z wyżej wymienionych powodów uznano, że komórki LSTM są lepszym rozwiązaniem w zadaniu związanym z generowaniem muzyki country.

Pracę można rozwijać na kilka różnych sposobów. Pierwszy z nich polega na dodaniu kolejnych instrumentów, tak aby model był w stanie obsługiwać całą gamę instrumentów.

Pozwoliłoby to uzyskać pełniejsze brzmienie, barwniejsze oraz znacznie bardziej złożone kompozycje muzyczne. Dodanie innych instrumentów nie jest zadaniem trudnym technicznie, ponieważ prawie cały proces przetwarzania dźwięków byłby taki sam. Jedynie należałoby zmienić sposób kodowania próbek, tak aby w sekwencji wynikowej można byłoby rozróżnić, którą nutę lub akord należy przypisać do określonego instrumentu. Jednakże głównym problemem jest znaczne powiększenie ilości próbek uczących, co wymaga większej mocy obliczeniowej oraz głębszej architektury sieci. Warto również wspomnieć, że należałoby wytworzyć nowe narzędzia do oceny jakości wieloinstrumentalnych utworów muzycznych. Kolejnym elementem, który mógłby zostać dodany to obsługa różnych czasów trwania dźwięków. W tym celu należałoby dodać kolejną klasę opisującą czas trwania dźwięku. W pracy każda nuta lub akord trwa taki sam okres czasu, natomiast w rzeczywistości dźwięki mają różny czas brzmienia. W analogiczny sposób można pomyśleć o uwzględnieniu w modelu przesunięć pomiędzy kolejnymi dźwiękami. Tutaj również wymagana byłaby kolejna klasa, która będzie opisywać wspomniane wcześniej przesunięcia. Zarówno w przypadku chęci dodania różnych przesunięć pomiędzy dźwiękami, jak i różnych czasów ich trwania zwiększa się ilość informacji, które model musi przetworzyć. Skutkuje to zwiększonym zapotrzebowaniem na moc obliczeniową oraz głębszą architekturę sieci.

Załącznik A

Do pracy załączono płytę DVD zawierającą w poszczególnych katalogach:

`/Praca_inzynierska.pdf` — wersja cyfrowa pracy,

`/Siec_neuronowa.zip` — archiwum zawierające dwie bazy utworów oraz wszystkie skrypty wykorzystane w projekcie.

Literatura

[1] Źródło utworów muzycznych z bazy treningowej. `https://freemidi.org/`.

[2] E. O. Agu, J. Z. Bako, M. A. Hambali. Analyzing Election Sentiments in Tweets with Gated Recurrent Units (GRU). *Asian Journal of Research in Computer Science*, 16(4):125–132, 2023.

[3] European Parliament. AI Act. `https://www.europarl.europa.eu/RegData/etudes/BRIE/2021/698792/EPRS_BRI(2021)698792_EN.pdf`.

[4] G. Neff, P. Nagy. Talking to Bots: Symbiotic Agency and the Case of Tay. *International Journal of Communication*, 10(4):4915–4931, 2016.

[5] G. Van Houdt, C. Mosquera, G. Nápoles. A Review on the Long Short-Term Memory Model. *Artificial Intelligence Review*, 53(1):5929–5955, 2020.

[6] N. Gupta. Artificial Neural Network. *Network and Complex Systems*, 3(1):24–28, 2013.

[7] R. Kline. Cybernetics, Automata Studies, and the Dartmouth Conference on Artificial Intelligence. *IEEE Annals of the History of Computing*, 33(4):5–16, 2011.

[8] B. Mahesh. Machine Learning Algorithms - A Review. *International Journal of Science and Research (IJSR)*, 9(1):381–386, 2020.

[9] J. H. Moor. An Analysis of the Turing Test. *Philosophical Studies: An International Journal for Philosophy in the Analytic Tradition*, 30(4):249–257, 1976.

[10] N. Helberger, N. Diakopoulos. ChatGPT and the AI Act. *Internet Policy Review*, 12(1):23–26, 2023.

[11] P. Hamet, J. Tremblay. Artificial intelligence in medicine. *Metabolism*, 69:36–40, 2017.

[12] P. P. Shinde, S. Shah. A Review of Machine Learning and Deep Learning Applications. *International Conference on Computing Communication Control and Automation*, strony 1–6, Pune, 2018.

[13] R. C. Schank. Where's the AI? *AI Magazine*, 12(4):38–49, 1991.

[14] D. M. Skapura. *Building Neural Networks*. Addison-Wesley Professional, Boston, 1995.

[15] Z. Stęgowski. Sztuczne Sieci Neuronowe. *Kernel.*, 1(1):16–19, 2004.