Jacek GREKOW

Politechnika Białostocka, Wydział Informatyki
ORCID: 0000-0003-2094-0107

doi:10.15199/48.2023.06.45

Generowanie wielogłosowej muzyki o określonej emocji przy użyciu wariacyjnego autoenkodera

Streszczenie. *Artykuł przedstawienia proces budowy modelu generującego wielogłosowe muzyczne sekwencje o określonej emocji. Opisano w nim proces przygotowania bazy przykładów uczących i budowę modelu generatywnego na bazie wariacyjnego autoenkodera. Przedstawiono eksperymenty implementacji warstw konwolucyjnych przeznaczonych do analizy wizualnej reprezentacji przykładów muzycznych. Wygenerowane pliki muzyczne poddano ewaluacji przez użycie metryk i porównanie ze zbiorem treningowym.*

Abstract*. This article presents the process of building a system generating polyphonic music content with a specified emotion. The process of preparing a training files and building a generative model based on a variational autoencoder was described. Experiments on the implementation of convolutional layers intended for analysis of the musical examples were presented. The generated examples were evaluated by using metrics and comparing them with the training set. (**Generating polyphonic music with a specified emotion using variational autoencoder**).*

Słowa kluczowe: generacja muzyki wielogłosowej, emocja w muzyce, wariacyjny autoenkoder, modele generatywne.
Keywords: polyphonic music generation, music emotion, variational autoencoder, generative models.

Wstęp

Generowanie muzyki przy użyciu modeli maszynowego uczenia jest nowym zjawiskiem wkraczającym na obszar kreatywnej twórczości człowieka. Jest coraz więcej systemów, które imitują ludzką kreatywność, uczą się na przykładach muzycznych stworzonych przez najwybitniejszych kompozytorów w dziejach muzyki [1].

Muzyka jest jedną z najbardziej abstrakcyjnych sztuk, wyraża myśli człowieka w formie zorganizowanych w czasie dźwięków. Analizując wysokości dźwięków utworu muzycznego wielogłosowego (polifonicznego), można je zwizualizować w postaci dwuwymiarowych graficznych obrazów [2], gdzie oś pozioma to czas, a pionowa to wysokości grających w danym czasie dźwięków. Ze względu na podobieństwo obrazów i muzyki, zagadnienia generowania obrazów i generowania muzyki mogą mieć podobne rozwiązania technologiczne. Inne metody wizualizacji muzyki przedstawiono w pracach [3, 4].

Jedną z głównych przyczyn dlaczego słuchamy muzyki jest odbiór emocji [5]. W zależności od zmieniających się w czasie parametrów takich jak melodia, barwa dźwięku, dynamika, rytm czy harmonia możemy odbierać różne emocje [6]. Dodanie elementu emocji do systemów generujących muzykę daje nam dodatkową kontrolę nad tworzonymi treściami. Podobne systemy mogą być wykorzystane w komunikacji maszyna-człowiek, w których oczekuje się nie tylko rozpoznania emocji, ale również wygenerowania odpowiedzi o właściwym zabarwieniu emocjonalnym.

Przegląd różnych zadań związanych z generowaniem muzyki z użyciem głębokiego uczenia przedstawiono w pracy [7]. Zaprezentowano w niej również stosowane reprezentacje muzyki, metody ewaluacji jak i popularne zbiory danych. Zauważono, że generowanie muzyki z określoną emocją jest jednym z przyszłościowych kierunków rozwoju badań. Inny przegląd prac poświęconych tej problematyce przedstawiono w [8].

Prace dotyczące generowania muzyki z emocją prezentują użycie różnych modeli głębokiego uczenia jak również różnych kategorii emocji. W podejściu przedstawionym w [9] zaprezentowano system generujący symboliczną muzykę przy użyciu sieci Biaxial LSTM. Zaproponowane rozwiązanie generuje polifoniczne przykłady z jedną z czterech podstawowych emocji. Podobny podział emocji podczas generowania monofonicznych sekwencji z użyciem wariacyjnego autoenkodera został zaprezentowany w pracy [10]. Idea generowania muzyki z określoną emocją (pozytywną/negatywną) z użyciem sieci LSTM przedstawiono w artykule [11]. Użycie Transformera do generowania muzyki o kontrolowanej emocji zostało zaproponowane w pracy [12]. Emocję opisano za pomocą trzech kategorii: negatywna, neutralna, pozytywna.

Celem tego artykułu jest przedstawienie procesu budowy modelu generującego wielogłosowe muzyczne sekwencje o określonej emocji. Zaprojektowany model powinien nauczyć się elementów muzycznych ze zbioru treningowego wpływających na emocję i zastosować je podczas generowania nowych przykładów.

Baza danych przykładów uczących

Pierwszą fazą budowy systemu do generowania muzyki jest budowa lub selekcja bazy danych z muzycznymi kompozycjami. W tej pracy użyto biblioteki music21 [13] zawierającej kompozycje Jana Sebastiana Bacha. W bibliotece treści utworów są zapisane w postaci symbolicznej, czyli mamy dostęp do takich parametrów dźwięków jak wysokość, długość, głośność, itp., i nie musimy ich dekodować z plików audio. Zbiór utworów zawiera przeważnie chorały (382) jak również inne kompozycje, w sumie razem 410 utworów. Pełna lista kompozycji w formacie MusicXML jest dostępna pod linkiem[1].

Aby użyć zbioru utworów music21 go generowania polifonicznych sekwencji poddano go kilku transformacjom. Pierwsze przetworzenie polegało na wyrównaniu czasu trwania nut. Na czas trwania nut w utworze wpływa zapisane w BPM (ang. Beats Per Minute) tempo utworu i rodzaj nuty. Ze względu na to, że utwory w bazie danych były zapisane w różnym tempie, tempa wszystkich utworów znormalizowano do 120 BPM, czyli 120 ćwierćnut na minutę. Wartości nut utworów o innym tempie niż 120 BPM zostały skorygowane. Otrzymano w ten sposób zbiór danych, w którym na długość trwania nut wpływają tylko ich rodzaje (nuty szesnastka, ósemka, ćwierćnuta, półnuta, cała nuta).

Druga transformacja zbioru danych polegała na ograniczeniu długości przykładu muzycznego do czterech taktów i selekcji utworów tylko o metrum 4/4, które są

[1] https://web.mit.edu/music21/doc/about/referenceCorpus.html

większością w music21. Spowodowała ona niewielką redukcję liczby przykładów w bazie danych. W ten sposób rytmiczna struktura przykładów została ujednolicona i ostatecznie zawierała 4 takty po 4 ćwierćnuty. W rezultacie otrzymano 8-sekundowe przykłady muzyczne, zawierające po 16 ćwierćnut każdy, w tempie 120 BPM.

Trzecia transformacja dotyczyła tonacji utworów, która jest różna w przykładach z bazy music21. Podczas generowania muzycznych sekwencji, najważniejsze są odległości między dźwiękami i ich wartości rytmiczne, tonacja nie odgrywa znaczącej roli. Ta sama melodia w różnych tonacjach brzmi podobnie, a zbiór treningowy w różnych tonacjach dodatkowo by utrudniał zadanie trenowania. Aby ułatwić trenowanie modelu wszystkie kompozycje zostały przetransponowane do tonacji C-dur lub c-moll. W ten sposób założono, że model będzie generował wielogłosowe sekwencje w skalach C-dur i c-moll. Po przeprowadzonych transformacjach otrzymano ujednolicony zbiór danych, 338 polifonicznych sekwencji o ujednoliconej długości (8 s.), tonacji C-dur lub c-moll. Wszystkie przetworzone przykłady zapisano w formacie MIDI.

Adnotacja przykładów etykietami emocji

Aby można było wytrenować model maszynowego uczenia generujący sekwencje muzyczne o określonej emocji potrzebne było oznaczenie przykładów muzycznych etykietami emocji. Podczas adnotacji użyto 4 podstawowych etykiet emocji: szczęście, złość, smutek, zadowolenie, odpowiadających 4 ćwiartkom modelu Russella [14] Q1-Q4. W modelu Russella emocje są rozłożone na płaszczyźnie podzielonej przez dwie prostopadłe osie: pobudzenie (ang. arousal) i walencja (ang. valence). Pobudzenie może być wysokie lub niskie, a walencja pozytywna lub negatywna. Użyte etykiety są oznaczeniem grupy emocji znajdujących się w danej ćwiartce, np. etykieta szczęście odnosi się grupy różnych emocji znajdujących się w ćwiartce Q1, w której pobudzenie jest wysokie, a walencja pozytywna. Podobny podział emocji na 4 podstawowe kategorie był użyty między innymi w pracach [9, 10, 15].

Oznaczane pliki muzyczne odtwarzane z formatu MIDI, były grane z jedną głośnością i jedną barwą dźwięku (MIDI instrument: Grand Piano), przez co głośność i barwa nie wpływały na emocje. To co wpływało na emocje w muzycznym fragmencie to wysokość, długość, rytmiczne ułożenie dźwięków, zależności harmoniczne między nimi, skala dur/moll.

Podczas adnotacji przykładów muzycznych można odnosić się do emocji odczuwanej lub zauważanej [16]. Emocja odczuwana to ta, którą słuchacz w danym momencie odczuwa, np. jeśli słucha czegoś bardzo smutnego, komunikuje, że emocja jest smutna i jednocześnie np. chce płakać. Emocja zauważana to ta, którą słuchacz dostrzega w utworze, ale fizycznie jej nie ulega, czyli np. słucha czegoś bardzo smutnego, komunikuje że emocja jest smutna, ale nie chce mu się płakać. W przeprowadzonym eksperymencie zadaniem muzycznych ekspertów było oznaczanie utworów MIDI emocją zauważaną.

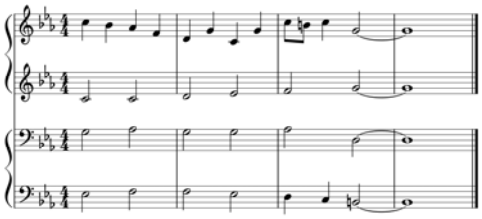
Adnotacja była wykonana przez trzech ekspertów z wyższym wykształceniem muzycznym. Opinie muzyczne ekspertów, ludzi którzy, grają w zespołach, komponują, interpretują muzykę na co dzień zajmują się są bardziej wiarygodne niż ludzi, którzy okazjonalnie zajmują się muzyką. Każdy z muzycznych ekspertów oznaczył wszystkie 338 pliki MIDI, przez co miał spojrzenie na cały zbiór muzyczny, jego odcienie emocjonalne, co nie zawsze jest zachowane przy oznaczaniu muzycznych baz danych. Oznaczenie całego zbioru przez każdego eksperta ma pozytywny efekt na jakość adnotacji, co zostało podkreślone w pracy [17]. Dane zebrane od trzech ekspertów zostały uśrednione. Rozważając zgodność odpowiedzi ekspertów obliczono współczynnik Alfa Cronbacha α = 0.88, co potwierdziło dobrą zgodność adnotacji. Liczby plików oznaczonych 4 emocjami przedstawiono w tabeli 1. Cały zbiór plików MIDI oznaczonych emocjami wraz z kodem poponowanego systemu można znaleźć pod linkiem[2].

Tabela 1. Liczby przykładów uczących oznaczonych 4 emocjami

Emocja	Skrót	Ćwiartka w modelu emocji / pobudzenie-walencja	Liczba
szczęście	e1	Q1 / wysokie-pozytywna	90
złość	e2	Q2 / wysokie-negatywna	89
smutek	e3	Q3 / niskie-negatywna	77
zadowolenie	e4	Q4 / niskie-pozytywna	82

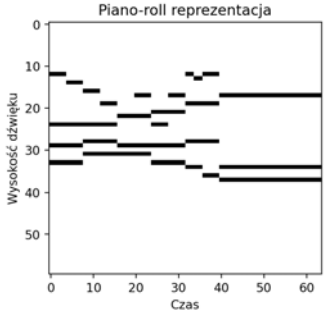
Kodowanie przykładów muzycznych

Dane z plików MIDI zanim będą użyte do trenowania modelu muszą zostać przetworzone, aby były zrozumiałe dla sieci neuronowej. Z tego powodu, że system generujący muzykę będzie się uczyć na utworach wielogłosowych, zdecydowano się na zakodowanie wszystkich plików MIDI z bazy danych przy użyciu piano-roll reprezentacji. W piano-roll reprezentacji oś pozioma opisuje znaczniki czasu trwania przykładu muzycznego, a oś pionowa wskazuje wysokości włączanych i wyłączanych dźwięków. Na rysunku 1 przedstawiono zapis nutowy fragmentu Chorału BWV 136 J.S. Bacha z bazy przykładów i odpowiadającą mu piano-roll reprezentację (rys. 2). Do odczytu plików MIDI i ich konwersji na piano-roll reprezentacje użyto narzędzia MusPy Toolkit [18].



Rys.1. Fragment zapisu nutowego Chorału BWV 136 J.S. Bacha

Piano-roll reprezentacja opisuje muzykę w macierzy czas-wysokość, gdzie kolumny są krokami czasowymi, a rzędy wysokościami dźwięków. Wartości w macierzy wskazują na obecność dźwięków w różnych krokach czasowych. Kształt standardowej macierzy to *T* × *128*, gdzie *T* jest numerem kroku czasowego. W formacie MIDI możliwe wysokości dźwięków (0-127), stąd liczba rzędów w macierzy wynosi 128.



Rys.2. Piano-roll reprezentacja fragmentu zapisu nutowego Chorału BWV 136 J.S. Bacha

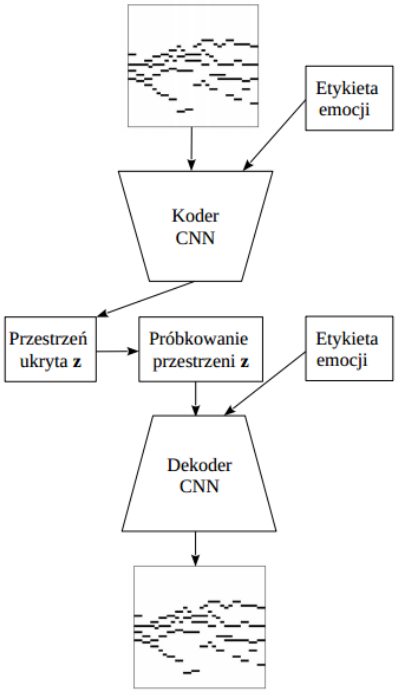
[2] https://github.com/grekowj/musgenvae_4v

Długość każdego przykładu z bazy danych odpowiada 4 taktom w metrum 4/4, co jest równe 4 ćwierćnutom na takt, i co daje w sumie 16 ćwierćnut. Najkrótsza nuta w bazie danych to nuta szesnastkowa i dlatego przykłady muzyczne zostały zakodowane (dyskretyzowane) krokiem czasowym odpowiadającym nucie szesnastkowej. Na każdą ćwierćnutę przypadają 4 nuty szesnastkowe, więc dzieląc cały przykład muzyczny najkrótszą nutą (szesnastką) otrzymujemy $T = 64$ kroki czasowe, *4 (takty) × 4 (ćwierćnuty) × 4 (szesnastki).*

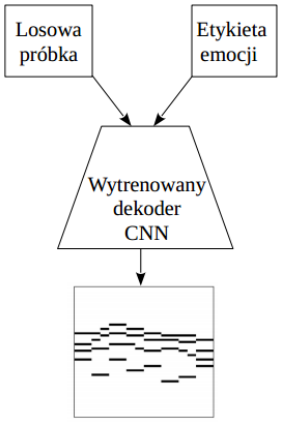
Po przeanalizowaniu wszystkich przykładów z plików MIDI okazało się, że bardzo wysokie i bardzo niskie dźwięki nie są wykorzystywane, umożliwiło to redukcję okna możliwych wysokości do 60 dźwięków. W rezultacie otrzymano kształt wyjściowego tensora reprezentującego przykład muzyczny *64 × 60* (*krok czasowy × wysokość dźwięku).* Przykładową wizualizację macierzy czas-wysokość pokazano na rysunku 2. Otrzymane macierze można interpretować jako wizualne reprezentacje przykładów muzycznych, czyli obrazy i użyć sieci neuronowych do przetwarzania obrazów.

Konstrukcja i implementacja systemu

Jako generatywny model został użyty warunkowy wariacyjny autoenkoder (CVAE - conditional variational autoencoder) [19]. Koduje on dane wejściowe w ukrytą przestrzeń o rozkładzie Gaussa, a następnie dekoduje próbki z ukrytej przestrzeni do formy danych podobnych, które były podane na wejściu (rys. 3). Właściwością wytrenowanego wariacyjnego autoenkodera jest to, że ukryta przestrzeń jest ciągła i można się po niej poruszać generując nowe dane. W CVAE mamy również na wejściu kodera i dekodera dodatkowy warunek (etykieta emocji), który umożliwia kontrolowane typu emocji generowanych przykładów muzycznych. Do generowania nowych przykładów jest używana tylko część dekodera (rys. 4). Na wejściu podawana jest etykieta emocji i próbka losowa o rozmiarze przestrzeni ukrytej.



Rys.3. Trenowanie modelu CVAE



Rys.4. Generowanie nowych przykładów z wytrenowanego dekodera

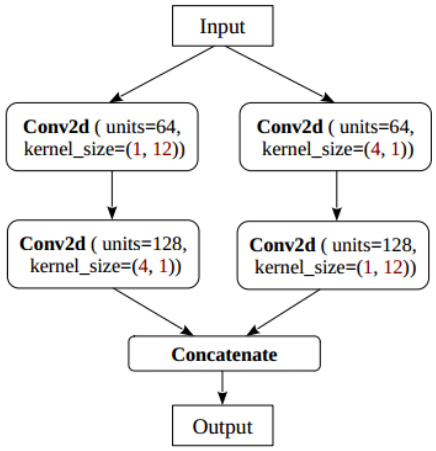
Podczas implementacji zaproponowano specjalne warstwy konwolucyjne do analizy wizualnej reprezentacji muzyki. Eksperymenty przeprowadzono na 2 modelach: bazowy i zaproponowany. Do implementacji w języku Python generatywnych modeli użyto biblioteki głębokiego uczenia Keras [20] i Tensorflow.

Kształt tensora reprezentacji muzycznych sekwencji na wejściu i wyjściu autoenkodera jest taki sam *(None, 64, 60, 1)*. Koder pobiera dane wejściowe x i określa średnią μ i odchylenie standardowe σ rozkładu Gaussa przestrzeni ukrytej z. Dekoder otrzymuje próbki z ukrytej przestrzeni z i rekonstruuje wejście x na wyjściu jako $\tilde{x}$. Funkcja straty sieci autoenkodera wariacyjnego jest sumą *straty rekonstrukcji* (S_R) i *straty ukrytej przestrzeni* (S_U). *Strata rekonstrukcji* wylicza różnicę między wejściem x i wyjściem $\tilde{x}$ używając entropii krzyżowej. *Strata ukrytej przestrzeni* (S_U) jest wyliczana używając dywergencji Kullbacka-Leiblera, która określa rozbieżność między docelowym rozkładem Gaussa i aktualnym rozkładem w ukrytej przestrzeni z.

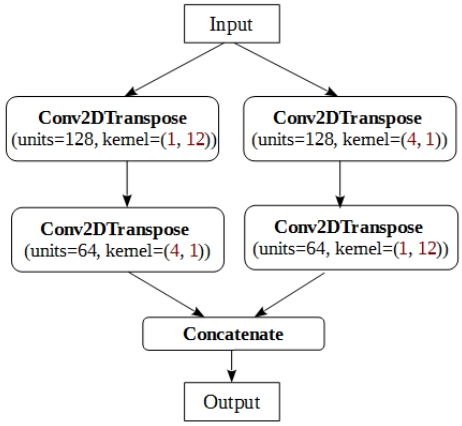
$$(1) \quad S_U = -\frac{1}{2}\sum_{i=1}^{K}(1+\log\sigma_i^2-\sigma_i^2-\mu_i^2)$$

gdzie K jest ilością wymiarów przestrzeni ukrytej z, μ i σ to średnia i odchylenie standardowe i wymiaru w przestrzeni ukrytej z.

Do analizy obrazów piano-roll reprezentacji zaproponowano specjalny rozkład warstw konwolucyjnych. Konstrukcja konwolucyjnej części kodera CNN została przedstawiona na rysunku 5. Składa się ona z dwóch równoległych ścieżek analizy, które są na końcu łączone warstwą łączącą (Concatenate). Każda ze ścieżek zawiera dwie sekwencyjne warstwy konwolucyjne (Conv2D) ze zwiększającą się liczbą jednostek (64, 128), i ze specjalnym rozmiarem jądra konwolucji (1,12) do analizy analizy 12 półtonów oktaw, i rozmiarem jądra (4, 1) do analizy kolejnych kroków czasowych. Na rysunku 6 przedstawiono dekoder, który analogicznie do kodera również składa się z dwóch równoległych ścieżek, sekwencyjnych warstw konwolucyjnych transponowanych (Conv2DTranspose), liczby jednostek konwolucyjnych się zmniejszają (64, 128), a rozmiary jąder konwolucji są analogiczne jak u kodera (1,12) i (4, 1). W standardowych warstwach konwolucyjnych najczęściej się używa jądra w formia kwadratu np. (3, 3) lub (5, 5). Z tego względu, że wymiary analizowanych obrazów mają określony sens (wysokość dźwięku, krok czasowy), zdecydowano się na modyfikację rozmiarów jąder konwolucji: jądro (1,12) analizuje całą oktawę (12 półtonów), a jądro (4, 1) analizuje 4 kroki czasowe równe jednej nucie ćwierćnuta, jednej czwartej taktu. Jako funkcje aktywacji w warstwach użyto funkcji ReLU.



Rys.5. Warstwy kodera CNN



Rys.6. Warstwy dekodera CNN

Zaproponowany CVAE z 2-ścieżkowym CNN w koderze i dekoderze, ze względu na to, że jest dedykowany do analizy reprezentacji muzycznych nazwano CVAE-Mus. Podczas przeprowadzanych eksperymentów porównano go z modelem bazowym CVAE-Base, który zamiast 2-ścieżkowych CNN zawierał sekwencyjnie połączone 2 warstwy konwolucyjne (Conv2D 256 i 128), o rozmiarze jądra (3,3) w koderze i analogicznie 2 warstwy konwolucyjne transponowane (Conv2DTranspose) w dekoderze. Przygotowane modele były trenowane optymalizatorem RMSprop (lr = 0,001) i 70 epokami. Ukryta przestrzeń była o rozmiarze 32.

W tabeli 2 przedstawiono otrzymane końcowe straty trenowania modeli. Zauważamy, że model CVAE-Mus lepiej się sprawdza w porównaniu z modelem bazowym, ma mniejszą stratą rekonstrukcji (126,15), jak również rozkład w ukrytej przestrzeni jest bardziej zbliżony do rozkładu Gaussa niż w przypadku modelu bazowego (mniejsza strata ukrytej przestrzeni).

Tabela 2. Straty rekonstrukcji i ukrytej przestrzeni dla CVAE

Model	Strata rekonstrukcji	Strata ukrytej przestrzeni
CVAE-Base	230,37	60,06
CVAE-Mus	126,15	57,29

Ewaluacja generowanych przykładów

Do ewaluacji wygenerowanych plików z określoną emocją użyto metryk [18], które analizują sekwencję muzyczną pod kątem wysokości dźwięków, ich ilości, użycia dźwięków z danej skali, itp. Wyliczono następujące metryki:

- *Zakres dźwięków* - współczynnik zdefiniowany jako różnica między największą i najmniejszą wysokością dźwięków użyta w sekwencji muzycznej;
- *Dźwięki w skali C-dur* - współczynnik zdefiniowany jako proporcja dźwięków z skali C-dur do wszystkich dźwięków w sekwencji muzycznej;
- *Dźwięki w skali c-moll* - współczynnik zdefiniowany jako proporcja dźwięków z skali c-moll do wszystkich dźwięków w sekwencji muzycznej;
- *Polifoniczność* - współczynnik zdefiniowany jako proporcja liczby kroków czasowych, w których mamy wielogłosowość do liczby wszystkich kroków czasowych w sekwencji muzycznej;

Aby ocenić wygenerowane przykłady muzyczne jako punkt odniesienia wybrano dane treningowe i z nimi porównywano wygenerowane przykłady. Przy użyciu dekodera z wytrenowanego modelu wygenerowano po 20 przykładów muzycznych dla każdej z czterech emocji emocji (e1, e2, e3, e4), czyli po 80 przykładów dla każdego z badanych modeli. Dla każdej wygenerowanej polifonicznej sekwencji, jak i dla każdego pliku ze zbioru treningowego obliczono 4 metryki. Tabela 3 przedstawia wyliczone średnie (μ) i odchylenia standardowe (σ) metryk otrzymanych dla muzyki z czterema emocjami (e1, e2, e3, e4) wygenerowanej przy użyciu zaproponowanego (CVAE-Mus) i bazowego modelu (CVAE-Base), i dla muzyki użytej do trenowania modeli. Pogrubioną czcionką zaznaczono wyniki z generowanych przykładów bliższe wynikom ze zbioru treningowego.

Zauważamy, że wartości średnie (μ) metryk otrzymanych dla proponowanego modelu są bliższe w większości przypadków (12/16) do metryk ze zbioru treningowego niż metryki z modelu bazowego. Zauważa się wyraźną poprawę metryk po zastosowaniu zaproponowanego modelu za wyjątkiem dwóch przypadków: *Dźwięki w skali C-dur* emocja e4 i *Dźwięki w skali c-moll* emocja e2.

Można zauważyć, że różnice między średnimi wartościami metryk dla poszczególnych emocji są mniejsze dla metryk *Dźwięki w skali C-dur*, *Dźwięki w skali c-moll* niż dla metryk *Zakres dźwięków* czy *Polifoniczność*. Można wnioskować, że model lepiej nauczył się korzystać z dźwięków dwóch przeciwstawnych skal dur i moll do zastosowania ich podczas generowania sekwencji muzycznych. Zazwyczaj skala dur jest kojarzona z emocjami pozytywnymi (e1, e4 - pozytywna walencja), a moll z negatywnymi (e2, e3 - negatywna walencja). Widać to w wyższych wartościach metryki *Dźwięki w skali C-dur* dla emocji e1 i e4, a mniejsze wartości dla e2 i e3. Odwrotne zachowanie jest metryki *Dźwięki w skali c-moll* - niższe wartości dla e1 i e4, a wyższe wartości dla e2 i e3. Można stwierdzić, że zaproponowany model lepiej rozpoznał wzorce w plikach oznaczonych emocjami niż model bazowy, przez co generuje pliki o charakterystykach bardziej zbliżonych do danych treningowych.

Podsumowanie

W artykule przedstawiono proces budowy modelu generującego wielogłosowe sekwencje muzyczne z wybraną emocją. Zbudowano bazę przykładów treningowych oznaczonych emocjami przez ekspertów muzycznych i zbudowano model na bazie warunkowego wariacyjnego autoenkodera.

Tabela 3. Średnie (μ) i odchylenia standardowe (σ) dla metryk otrzymanych dla przykładów muzycznych generowanych i treningowych, oznaczonych czterema emocjami (e1-e4)

Metryka	Emocja	Zbiór wygenerowany modelem CVAE-Base μ (σ)	Zbiór wygenerowany modelem CVAE-Mus μ (σ)	Zbiór treningowy μ (σ)
Zakres dźwięków	e1	39,45 (8,33)	**36,55** (8,32)	31,71 (3,65)
	e2	34,50 (6,70)	**33,60** (7,95)	30,70 (2,57)
	e3	38,25 (5,97)	**37,20** (11,92)	27,73 (3,25)
	e4	4,.35 (7,54)	**36,20** (11,28)	27,04 (4,96)
Dźwięki w skali C-dur	e1	0,99 (0,02)	**0,97** (0,03)	0,96 (0,04)
	e2	0,48 (0,09)	**0,68** (0,11)	0,68 (0,09)
	e3	0,50 (0,10)	**0,66** (0,10)	0,70 (0,10)
	e4	**0,97** (0,04)	0,96 (0,06)	0,98 (0,03)
Dźwięki w skali c-moll	e1	0,52 (0,06)	**0,65** (0,08)	0,61 (0,06)
	e2	**0,96** (0,05)	0,97 (0,03)	0,89 (0,08)
	e3	0,92 (0,06)	**0,91** (0,08)	0,88 (0,09)
	e4	0,57 (0,07)	**0,61** (0,06)	0,63 (0,06)
Polifoniczność	e1	0,35 (0,17)	**0,59** (0,22)	0,99 (0,05)
	e2	0,46 (0,11)	**0,58** (0,18)	0,99 (0,03)
	e3	0,56 (0,15)	**0,65** (0,25)	1,00 (0,01)
	e4	0,61 (0,18)	**0,77** (0,13)	0,97 (0,11)

Zaproponowano specjalną strukturę dla warstw konwolucyjnych w koderze i dekoderze CVAE do kodowania i dekodowania wizualnych reprezentacji przykładów muzycznych. Zaprezentowana w artykule oryginalna konstrukcja warstw konwolucyjnych do analizy polifonicznych przykładów muzycznych wykazała przewagę nad sekwencyjną strukturą standardowych warstw konwolucyjnych. Wytrenowany model rozpoznał wzorce wpływające na emocji w zbiorze treningowym i był w stanie je zastosować w generowanych przykładach.

Ewaluacja wygenerowanych przykładów muzycznych wykazała, że sekwencje otrzymane za pomocą zaproponowanego modelu są bliższe przykładom ze zbioru treningowego niż sekwencje generowane z użyciem modelu bazowego. Z powodu elementu losowego jakim jest użycie wektora wartości losowych w przestrzeni ukrytej, generowane przykłady różnią się od zbioru treningowego jednak ich emocjonalne charakterystyki są zbliżone do danych uczących.

Ograniczenia przedstawionego rozwiązania to na pewno niewielka długość generowanych sekwencji. Również większa liczba przykładów uczących związana z kosztownym procesem adnotacji mogłaby polepszyć otrzymane rezultaty.

W przyszłości można by przebadać jeszcze inne warianty struktury połączenia warstw konwolucyjnych do badania zakodowanych reprezentacji polifonicznej muzyki. Zaprezentowane podejście nie rozwiązuje całkowicie problemu, a wręcz wskazuje kierunki dalszego badania generowania emocjonalnej muzyki polifonicznej z użyciem generatywnych modeli.

Zrealizowano w ramach Pracy Własnej Zespołowej Katedry Systemów Informacyjnych i Sieci Komputerowych Wydziału Informatyki Politechnice Białostockiej. Środki na rok 2023.

***Autorzy**: dr hab. inż. Jacek Grekow, Politechnika Białostocka, Wydział Informatyki, Wiejska 45A, Białystok 15-351, E-mail: j.grekow@pb.edu.pl.*

LITERATURA

[1] Briot J.-P., From artificial neural networks to deep learning for music generation: history, concepts and trends, *Neural. Comput. Appl.*, vol. 33 (2021), 39–65

[2] Dong H.-W., Hsiao W.-Y., Yang Y.-H., Pypianoroll: Open source python package for handling multitrack pianorolls*, 19th Int. Soc. Music Inf. Retr. Conf. (ISMIR)*, (2018), 101–108

[3] Grekow J., Metoda transformowania muzyki w figury 4D, *Przegląd Elektrotechniczny*, 91 (2015), nr 4, 159-162

[4] M. Kania, D. Kania, Trajektoria kwintowa – dwuwymiarowa reprezentacja muzyki, Przegląd Elektrotechniczny, 98 (2022), nr 6, 70-73

[5] Pratt C. C., Music as the language of emotion, *Washington, U.S. Govt. Print. Off.: The Library of Congress*, (1950)

[6] Grekow J., From Content-based Music Emotion Recognition to Emotion Maps of Musical Pieces, *Stud. Comput. Intell. Springer*, vol. 747 (2018)

[7] Ji S., Luo J., Yang X., A comprehensive survey on deep music generation: Multi-level representations, algorithms, evaluations, and future directions, *CoRR*, (2020)

[8] Zhao Z., Liu H., Li S., Pang J., Zhang M., Qin Y., Wang L., Wu Q., A review of intelligent music generation systems, *arXiv*, (2022)

[9] Zhao K., Li S., Cai J., Wang H., Wang J., An emotional symbolic music generation system based on LSTM networks, *IEEE 3rd Info., Technol., Networking, Electr. Automat. Contr. Conf. (ITNEC),* (2019), 2039–2043

[10] Grekow J., Dimitrova-Grekow T., Monophonic music generation with a given emotion using conditional variational autoencoder, *IEEE Access*, vol. 9 (2021), 129088–129101

[11] Ferreira L., Whitehead J., Learning to generate music with sentiment, *20th Int. Soc. Music Inf. Retr. Conf. (ISMIR)*, (2019), 384–390

[12] Pangestu M. A., Suyanto S., Generating music with emotion using transformer, *Int. Conf. Comput. Sci. Eng. (IC2SE*), (2021), 1–6

[13] Cuthbert M., Ariza C., Music21: A toolkit for computer-aided musicology and symbolic music data., *11th Int. Soc. Music Inf. Retr. Conf. (ISMIR),* (2010), 637–642

[14] Russell J. A., A circumplex model of affect, *J. Pers. Soc. Psychol.*, vol. 39 (1980), no. 6, 1161–1178

[15] Wang X., Xiaoou C., Yang D., Wu Y., Music emotion classification of chinese songs based on lyrics using tf*idf and rhyme., 12th Int. Soc. Music Inf. Retr. Conf. (ISMIR), (2011), 765–770

[16] Gabrielsson A., Emotion perceived and emotion felt: Same or different?, *Music. Sci.*, vol. 5 (2002), no. 1 suppl, 123–147

[17] Aljanaki A., Yang Y.-H., Soleymani M., Developing a benchmark for emotional analysis of music, PLOS ONE, vol. 12 (2017), no. 3, 1–22

[18] Dong H.-W., Chen K., McAuley J., Berg-Kirkpatrick T., Muspy: A toolkit for symbolic music generation, *21st Int. Soc. Music Inf. Retr. Conf. (ISMIR*), (2020)

[19] Sohn K., Yan X., Lee H., Learning structured output representation using deep conditional generative models, *28th Int. Conf. Neural Inf. Process. Syst.,* (2015), 3483–3491

[20] Chollet F., et al., Keras, https://keras.io, (2015)