Jacek GREKOW

ORCID: 0000-0003-2094-0107

OPEN ACCESS

DOI: 10.15199/48.2025.04.33

Regulacja przestrzeni ukrytej wariacyjnego autoenkodera względem emocji generowanych sekwencji muzycznych

Regulation of the latent space of a variational autoencoder with respect to the emotions of generated music sequences

Streszczenie. *Artykuł przedstawienia nową regulację przestrzeni ukrytej wariacyjnego autoenkodera w celu jej powiązania z emocją generowanych przykładów muzycznych. Jako model emocji użyto dwuwymiarowy model Russella, którego 4 ćwiartki odpowiadają podstawowym emocjom, jak szczęście, złość, smutek i zadowolenie. Zbudowano modele wariacyjnego autoenkodera, wykorzystujące rekurencyjne sieci neuronowe, które uczono na zbiorze jednogłosowych muzycznych sekwencji oznaczonych emocjami. Ewaluacji poddano otrzymaną przestrzeń ukrytą, jak i wygenerowane pliki muzyczne o różnych emocjach.*

Abstract*. The article presents a new regulation of the latent space of a variational autoencoder in order to connect it with the emotion of generated music examples. A two-dimensional Russell model was used as the emotion model, with its 4 quadrants corresponding to basic emotions such as happiness, anger, sadness, and relaxation. Variational autoencoder models employing recurrent neural networks were constructed and trained on a dataset of monophonic music sequences labeled with emotions. The obtained latent space, as well as the generated music files with different emotions, were evaluated.*

Słowa kluczowe: regulacja przestrzeni ukrytej, wariacyjny autoenkoder, generacja muzyki, emocja w muzyce
Keywords: latent space regulation, variational autoencoder, music generation, music emotion

Wstęp

Generowanie muzyki przy użyciu modeli głębokiego uczenia jest nowym pasjonującym zjawiskiem konkurującym z kreatywną twórczością człowieka. Jest coraz więcej systemów, które tworzą muzykę i trenowane są na przykładach muzycznych stworzonych przez kompozytorów poprzednich epok, jak i artystów współczesnych [1], [2]. Wśród modeli generatywnych często wybierany jest wariacyjny autoencoder (VAE) [3] ze względu na możliwość kontrolowania cech generowanych danych. Został on między innymi wykorzystany w pracy [4] gdzie przestrzeń ukrytą powiązano ze stylem generowanej muzyki symbolicznej. W [5] model VAE kontroluje tonalne napięcie generowanej muzyki. Generowana muzyka jest podobna do muzyki wejściowej, ale poprzez manipulowanie rytmem i wysokością dźwięków model wpływa na napięcie tonalne.

Generowanie muzyki o określonej emocji jest jednym z przyszłościowych kierunków rozwoju badań, co zostało zauważone w [6]. W podejściu przedstawionym w [7] zaprezentowano system generujący symboliczną muzykę przy użyciu sieci LSTM. Zaproponowane rozwiązanie generuje polifoniczne przykłady z jedną z czterech podstawowych emocji. Również prace [8] i [9] wykorzystują generatywny model VAE odpowiednio z sieciami rekurencyjnymi i konwolucyjnymi do generowania muzyki o określonej kategorii emocji. Autorzy pracy [10] zaproponowali model bazujący na jednostkach rekurencyjnych i hybrydowym mechanizmie nagradzania do generowanie muzyki o określonej emocji. Użyto cech takich jak histogramy wysokości nut do reprezentacji emocji. W pracach [7], [8], [9], [10] użyto etykiet emocji odpowiadających 4 ćwiartkom modelu Russella [11] co wskazuje na prostotę (tylko 4 kategorii emocji) i popularność tego modelu. Zadaniem odwrotnym do generowania muzyki o określonej emocji jest detekcja emocji w plikach muzycznych, która została opisana między innymi w [12].

Ukryta przestrzeń wariacyjnego autoenkodera może być powiązana z różnymi cechami generowanych sekwencji. W pracy [13] zaproponowano regulację, która łączyła liczbę granych nut sekwencji muzycznej z jedną osią przestrzeni ukrytej. Zademonstrowano zastosowaną regulację do generowania wariantów podanej melodii. Praca [14] przedstawia powiązanie cech generowanych obiektów, obrazów i sekwencji muzycznych, z poszczególnymi wymiarami ukrytej przestrzeni. Zaproponowana regulacja tworzy 2 macierze odległości: cech przykładów treningowych i przestrzeni ukrytej, między którymi wyliczany jest średni błąd bezwzględny, uwzględniany w całkowitej stracie modelu VAE. Generowane przykłady muzyczne są losowane z przestrzeni ukrytej powiązanej z cechami jak złożoność rytmiczna, zakres dźwięków, gęstość nut i kształt melodii. W [15] wykorzystano regulację przestrzeni ukrytej bazując na rozwiązaniu przedstawionym w pracy [14] zmieniając tylko wartości cech sekwencji muzycznych z ciągłych na dyskretne. Model wariacyjnego autoenkodera wykorzystano do generowania symbolicznej muzyki o 4 podstawowych emocjach.

Celem tego artykułu jest zaprezentowanie nowej regulacji przestrzeni ukrytej wariacyjnego autoenkodera, która rozmieści obszary dwuwymiarowej przestrzeni wg ćwiartek modelu emocji. Poprzez odwzorowanie modelu emocji w przestrzeni ukrytej, przestrzeń ta stanie się bardziej interpretowalna podczas generowania nowych przykładów. Generowanie polega na losowym próbkowaniu z przestrzeni ukrytej i budowaniu sekwencji przez dekoder modelu. Zmodyfikowany model wariacyjnego autoenkodera zostanie użyty do generowania monofonicznych sekwencji muzycznych o określonej emocji.

Zbiór przykładów uczących

Do trenowania modelu generującego użyto zbioru przykładów uczących w formacie MIDI oznaczonych emocjami, który został stworzony w pracy [8]. Jest on publicznie dostępny pod linkiem.[1] W zbiorze tym znajdują się przetworzone fragmenty kompozycji Jana Sebastiana Bacha pobrane z biblioteki music21 [16].

Zanim odczytane pliki MIDI z biblioteki music21 stały się danymi uczącymi dla sieci neuronowej poddano je kilku transformacjom. Pierwsze przetworzenie polegało na wyrównaniu czasu trwania nut. Tempo wszystkich utworów

[1] https://github.com/grekowj/musgenvae

zostało wyrównane do 120 BPM (ang. Beats Per Minute). Otrzymano w ten sposób zbiór danych, w którym na długość trwania nut wpływają tylko ich rodzaje (nuty szesnastka, ósemka, ćwierćnuta, półnuta, cała nuta).

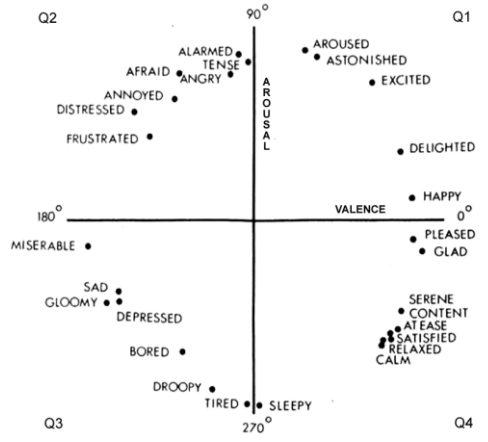
Druga transformacja zbioru danych polegała na selekcji utworów tylko o metrum 4/4 i ograniczeniu długości przykładu muzycznego do czterech taktów. W ten sposób rytmiczna struktura przykładów uczących została ujednolicona: cztery takty utworu o metrum 4/4. Z tego względu, że ujednolicone tempo wynosiło 120 BPM otrzymano 8 s. przykłady muzyczne, o długości odpowiadającej 16 ćwierćnutom.

Trzecia transformacja dotyczyła ujednolicenia tonacji utworów. Podczas generowania muzycznych sekwencji, najważniejsze są relacje i odległości między dźwiękami i ich wartości rytmiczne, tonacja nie odgrywa znaczącej roli. Aby ułatwić trenowanie modelu wariacyjnego autoencodera wszystkie kompozycje zostały przetransponowane do tonacji C-dur lub c-moll.

Celem naszego modelu jest generowanie jednogłosowych sekwencji muzycznych i dlatego ostatnią transformacją było pobranie tylko najwyższego głosu, sopranu wielogłosowej kompozycji, która zazwyczaj zawiera główną melodię utworu.

Po przetworzeniu odczytanych danych otrzymano ujednolicony zbiór, 334 monofonicznych sekwencji o ujednoliconej długości (8 s.), tonacji C-dur lub c-moll. Wszystkie przetworzone przykłady zapisano w formacie MIDI.

Zestaw przykładów uczących z pracy [8] jest oznaczony 4 podstawowymi etykietami emocjami: szczęście, złość, smutek, zadowolenie, odpowiadającymi 4 ćwiartkom modelu Russella [11] Q1-Q4 (rys. 1). 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. Same etykiety są oznaczeniem grupy emocji znajdujących się w danej ćwiartce, np. etykieta szczęście odnosi się grupy różnych emocji znajdujących się w ćwiartce Q1 gdzie pobudzenie jest wysokie, a walencja pozytywna. Ilości plików oznaczonych 4 emocjami przedstawiono w Tabeli 1.



Rys. 1. Model emocji Russella [11]

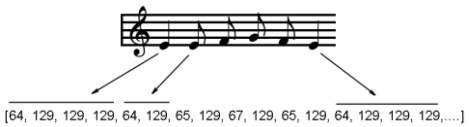
Tabela 1. Ilości plików przykładów uczących oznaczonych 4 podstawowymi emocjami

Emocja	Skrót	Ćwiartka w modelu emocji / pobudzenie-walencja	Liczba
szczęście	e1	Q1 / wysokie-pozytywna	80
złość	e2	Q2 / wysokie-negatywna	79
smutek	e3	Q3 / niskie-negatywna	93
zadowolenie	e4	Q4 / niskie-pozytywna	92

Kodowanie monofonicznych sekwencji

Dane z plików MIDI zanim będą użyte do trenowania modelu muszą zostać przetworzone, aby były zrozumiałe dla sieci neuronowej. Z tego względu, że model będzie się uczył na monofonicznych sekwencjach, pliki MIDI zakodowano przy użyciu pitch-based reprezentacji. Konwersji dokonano za pomocą biblioteki MusPy Toolkit [17]. Pitch-based reprezentacja koduje muzyczną sekwencję w tokeny reprezentujące wysokość, pauzę czy podtrzymanie poprzedniej wartości. Wyjściowy kształt sekwencji jest *T* × *1* gdzie *T* jest liczbą kroków czasowych. Wartości tokenów w sekwencji wskazują czy w danym kroku czasowym mamy dźwięk (0-127), pauzę (128), czy podtrzymanie poprzedniej wartości (129). Podtrzymanie poprzedniej wartości używane jest, gdy nuta jest dłuższa niż rozdzielczość kodowania. W naszym przypadku najkrótsza nuta w zbiorze uczącym to nuta szesnastkowa, więc i rozdzielczość kodowania tej samej wartości. Szczegółowy przykład transformacji został przedstawiony na rysunku 2, gdzie pierwsza ćwierćnuta została zakodowana czterema wartościami: wysokość nuty E4 (64) i trzema tokenami podtrzymania poprzedniej wartości (129).

Czterotaktowe sekwencje o metrum 4/4 zostały zakodowane przy użyciu pitch-based reprezentacji i rozdzielczości kodowania równej nucie szesnastkowej, w wyniku czego otrzymano 64 kroki czasowe. Dodatkowo aby ułatwić uczenie sieci neuronowej, sprawdzono liczbę różnych wysokości nut w zbiorze uczącym i ilość tokenów zredukowano do 29. Po dodaniu tokenu pauzy i podtrzymania uzyskano 31 różnych wartości, które zakodowano „gorącą jedynką". Kształt tensora danych jednego przykładu uczącego był *64* × *31* (*kroki* × *tokeny*).



Rys. 2. Przykład tworzenia pitch-based reprezentacji sekwencji muzycznej

Regulacja przestrzeni ukrytej

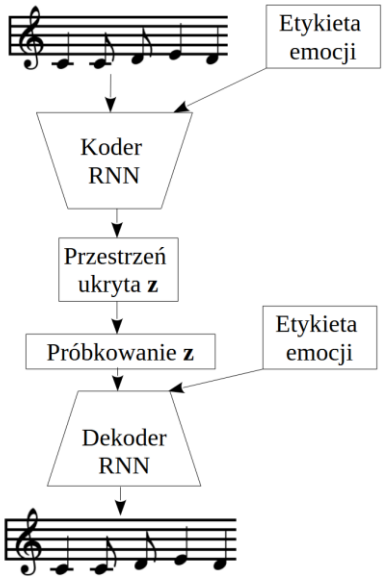
Celem nowej regulacji wariacyjnego autoenkodera jest powiązanie modelu emocji z ćwiartkami przestrzeni ukrytej, wykorzystywanej do generowanych sekwencji muzycznych. Jako generatywny model został użyty warunkowy wariacyjny autoenkoder (CVAE - conditional variational autoencoder) [18]. Koduje on dane wejściowe, w naszym przypadku sekwencje muzyczne, w ukrytą przestrzeń o rozkładzie Gaussa, a następnie dekoduje próbki z ukrytej przestrzeni do formy danych, które były podane na wejściu (rys. 3). Umożliwia on generowanie muzycznych sekwencji o określonej emocji poprzez losowe próbkowanie z przestrzeni ukrytej. W CVAE mamy na wejściu kodera i dekodera dodatkowy warunek (etykieta emocji), który umożliwia kontrolowanie typu emocji generowanych przykładów muzycznych.

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 standardowego autoenkodera wariacyjnego [3] jest sumą *straty rekonstrukcji* L_R i *straty regulacji* L_{KLD}.

$$L_U = L_R + L_{KLD} \tag{1}$$

Strata rekonstrukcji L_R wylicza różnicę między wejściem x i wyjściem $\tilde{x}$.

(2) $$L_R = \frac{1}{N}\sum_{j=1}^{N}\left(\tilde{x}_j - x_j\right)^2$$



Rys. 3. Trenowanie modelu CVAE

Strata regulacji L_{KLD} reguluje przestrzeń ukrytą i jest wyliczana używając dywergencji Kullbacka-Leiblera, która określa rozbieżność między docelowym rozkładem Gaussa i aktualnym rozkładem w ukrytej przestrzeni z. Strata ta zapewnia, że pod wpływem treningu sieci rozkład punktów w przestrzeni ukrytej zbliża się do rozkładu Gaussa.

(3) $$L_{KLD} = -\frac{1}{2}\sum_{i=1}^{K}\left(1 + \log\sigma_i^2 - \sigma_i^2 - \mu_i^2\right)$$

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

Aby wprowadzić uporządkowanie rozłożenia punktów w przestrzeni ukrytej adekwatnie do 4-ćwiartkowego modelu emocji wprowadzono *stratę emocji*. Do jej obliczenia wykorzystano etykiety emocji (pobudzenie A, i walencję V) próbek we wsadzie danych (ang. batch) podawanych jednocześnie do sieci podczas treningu, oraz wartości aktualnych pozycji próbek treningowych w przestrzeni ukrytej z, w naszym przypadku o kształcie równym 2.

Stratę emocji wyliczono w 2 wariantach: stratę dla pobudzenia L_{EmoA}, czyli dla punktów na osi pionowej ukrytej przestrzeni i stratę dla walencji L_{EmoV}, czyli dla punktów na osi poziomej.

(4) $$L_{EmoA} = -\text{CosineSimilarity}\left(A, z(1)\right)$$

gdzie A jest wektorem wartości pobudzenia (wysokie-niskie) przykładów treningowych znajdujących się w danym wsadzie, zależnym od wartości etykiety emocji. $z(1)$ jest wektorem wartości aktualnej pozycji punktów na osi pionowej przestrzeni ukrytej.

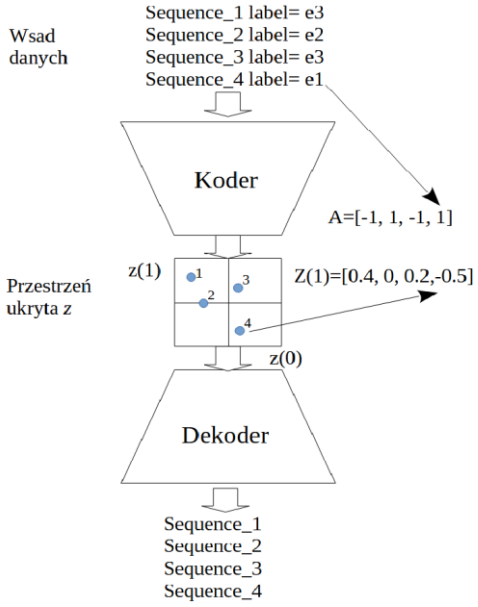
(5) $$A(e)_j = \begin{cases} 1 & dla \quad e = e1 \vee e = e2 \\ -1 & dla \quad e = e3 \vee e = e4 \end{cases}$$

(6) $$L_{EmoV} = -\text{CosineSimilarity}\left(V, z(0)\right)$$

gdzie V jest wektorem wartości walencji (pozytywna-negatywna) przykładów treningowych znajdujących się w danym wsadzie, zależnym od wartości etykiety emocji. $z(0)$ jest wektorem wartości aktualnej pozycji punktów na osi poziomej przestrzeni ukrytej.

(7) $$V(e)_j = \begin{cases} 1 & dla \quad e = e1 \vee e = e4 \\ -1 & dla \quad e = e2 \vee e = e3 \end{cases}$$

Na rysunku 4 przedstawiono budowę wektora A zawierającego wartości pobudzenia wsadu treningowego i wektora $z(1)$ reprezentującego wartości aktualnej pozycji punktów na osi pionowej przestrzeni ukrytej. Przykłady treningowe z wsadu danych podawane są do kodera i odwzorowywane w postaci punktów na dwuwymiarowej przestrzeni ukrytej. Na postawie wartości etykiet wsadu budowany jest wektor A, a pozycje punktów na osi pionowej wyznaczają wartości wektora $z(1)$. Analogicznie są tworzone V i $z(0)$.



Rys. 4. Budowa wektorów A i $z(1)$ wykorzystywanych do wyliczenia *straty emo*cji L_{EmoA}

Do wyliczenia straty użyto podobieństwa kosinusowego, które mierzy kąt między zbudowanymi wektorami i zwraca wartości z przedziału $[-1, 1]$, gdzie 1 odpowiada kątowi 0°, a -1 kątowi 180°. Jeśli kąt między wektorem emocji przykładów treningowych w danym wsadzie i wektorem wartości aktualnej pozycji punktu w przestrzeni ukrytej jest duży, to strata jest większa, czyli nie ma zbieżności między wartościami emocji i położeniem punktów w przestrzeni ukrytej. Jeśli kąt między wektorami jest mniejszy, strata się zmniejsza, czyli jest zbieżność między wartościami emocji i położeniem punktów w przestrzeni ukrytej.

Celem *straty emocji* jest nakłonienie przestrzeni ukrytej do umieszczenia punktów we właściwych półkulach. Strata L_{EmoA} przesuwa punkty pomiędzy górną i dolną półkulą, a L_{EmoV} pomiędzy lewą i prawą. Użycie tych dwóch start wpływa na umieszczenie punktów we właściwych ćwiartkach. Ostatecznie stratę wyliczano poprzez dodanie L_{EmoA} i L_{EmoV} do straty standardowego autoenkodera wariacyjnego.

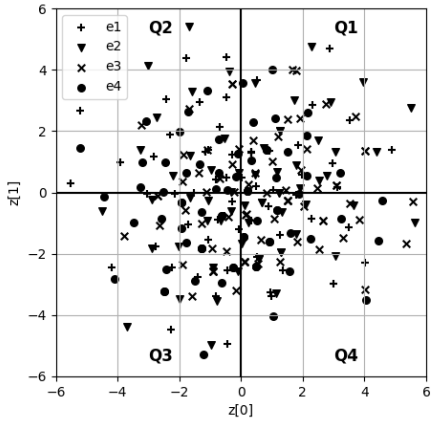
(8) $$L_U = L_R + L_{KLD} + L_{EmoA} + L_{EmoV}$$

Implementacja i trenowanie modeli

Do sprawdzenia zastosowanej regulacji zbudowano 2 modele CVAE, jeden bez regulacji CVAE-Base i drugi z nową regulacją CVAE-EmoReg. Koder i dekoder modeli został zaimplementowany przy użyciu rekurencyjnej sieci neuronowej (RNN), składającej się z 512 jednostek GRU (ang. Gated Recurrent Units). Kształt zakodowanych pitch-based reprezentacją muzycznych sekwencji był na wejściu i wyjściu modelu taki sam *(None, 64, 31).* Model CVAE-Base został opisany szczegółowo w pracy [8], gdzie był użyty do generowania sekwencji muzycznych o określonej emocji, a wybór emocji był określany dodatkowym warunkiem wejściowym. W tej pracy został on rozszerzony o nową regulację przestrzeni ukrytej. Przestrzeń ukryta została powiązana z dwuwymiarowym modelem emocji i dlatego posiada 2 wymiary. Przygotowane modele były trenowane optymalizatorem RMSprop (*lr* = 0.001), 700 epokami i przy wsadzie o rozmiarze równym 16. Generatywne modele zostały zaimplementowane w języku Python z użyciem biblioteki głębokiego uczenia Keras [19] i Tensorflow.

Ewaluacja regulacji przestrzeni ukrytej VAE

Rysunek 5 przedstawia przestrzeń ukrytą modelu bazowego CVAE-Base, w którym zauważamy brak uporządkowania punktów reprezentujących dane treningowe z etykietami e1, e2, e3, e4 w ćwiartkach modelu emocji.



Rys. 5. Przestrzeń ukryta modelu bazowego CVAE-Base

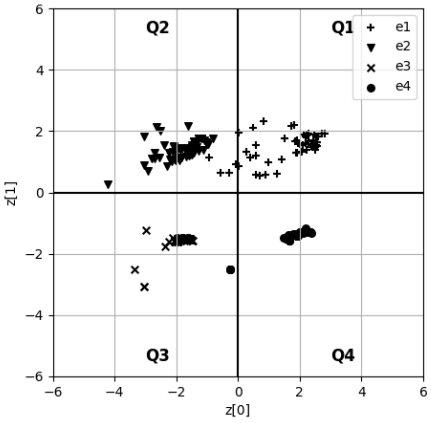
Na rysunkach 6 i 7 przedstawiono przestrzeń ukrytą modelu CVAE-EmoReg, w którym widzimy wpływ regulacji na uporządkowanie rozmieszczenia zbioru treningowego w poszczególnych ćwiartkach. Pierwszy z nich (rys. 6) przedstawia stan podczas początku trenowania (50 epoka), w którym punkty są już w odpowiednich ćwiartkach, ale brak ciągłości w rozłożeniu przykładów, co kontrastuje z przestrzenią ukrytą po wytrenowaniu modelu, gdzie mamy równomiernie rozłożone przykłady we wszystkich ćwiartkach.

Celem regulacji przestrzeni ukrytej było rozłożenie punktów reprezentujących dane treningowe w ćwiartkach odpowiadających ćwiartkom modelu emocji. Do jej ewaluacji wykorzystano metrykę dokładności rozmieszczenia punktów w ćwiartkach przestrzeni ukrytej.

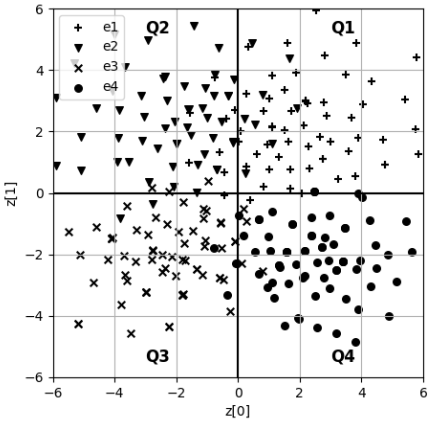
(9) $$D_i = \frac{\mathrm{PQ_i}}{\mathrm{PT_i}} \times 100\%$$

gdzie $i = \{1, 2, 3, 4\}$ jest numerem ćwiartki modelu emocji, PQ ilością punktów znajdujących się w danej ćwiartce, a PT ilością przykładów treningowych reprezentujących daną emocję i.

Tabela 2 przedstawia dokładności rozmieszczenie punktów D_i w poszczególnych ćwiartkach. 100% oznaczałoby, że wszystkie przykłady treningowe o danej emocji zostały umieszczone we właściwej ćwiartce.



Rys. 6. Przestrzeń ukryta modelu CVAE-EmoReg w trakcie trenowania przy 50 epoce



Rys. 7. Przestrzeń ukryta modelu CVAE-EmoReg po zakończeniu trenowania

Tabela 2. Dokładność rozmieszczenia D_i punktów w ćwiartkach przestrzeni ukrytej modelu pobudzenie-walencja

Ćwiartka modelu	Model CVAE-Base	Model CVAE-EmoReg
Q1	24%	83%
Q2	20%	84%
Q3	23%	92%
Q4	25%	92%
Średnia	23%	88%

Widzimy wysokie wartości dokładności rozmieszczenia punktów dla poszczególnych emocji (83-92%) dla modelu CVAE-EmoReg, które potwierdzają działanie regulacji w przestrzeni ukrytej. Dla modelu bazowego CVAE-Base, gdzie brak regulacji przestrzeni ukrytej, wartość średnia dokładności to zaledwie 23% w porównaniu do 88% zaproponowanego modelu CVAE-EmoReg.

Przykłady wygenerowanych sekwencji muzycznych

Na rysunku 8 przedstawiono przykłady wygenerowanych muzycznych sekwencji o zadanej emocji przy użyciu dekodera modelu CVAE-EmoReg. Z zapisu

nutowego na rysunku 8a i 8b widzimy większą gęstość nut na przestrzeni czterech taktów, co jest związane emocjami e1 i e2 (wysokie pobudzenie). Przykłady o niskim pobudzeniu, mniej nut, są zaprezentowane na rysunku 8c i 8d, co odpowiada emocjom e3, e4. Użycie dźwięków skali C-dur, co wiąże się z pozytywną walencją, emocje e1 i e4, widzimy na rysunku 8a i 8d. Dźwięki skali c-moll, wiążące się z negatywną walencją, widzimy na rysunku 8b i 8c, które reprezentują emocje e2 i e3.

Rys. 8. Przykłady wygenerowanych muzycznych sekwencji o zadanej emocji (a) e1, (b) e2, (c) e3, i (d) e4

Ewaluacja wygenerowanych przykładów

Do ewaluacji wygenerowanych plików z określoną emocją użyto metryk z biblioteki MusPy [17], które analizują pliki MIDI pod kątem wysokości dźwięków, ich ilości i użycia danej skali. Wyliczono następujące metryki:

- *Zakres dźwięków* - współczynnik zdefiniowany jako różnica między największą i najmniejszą wysokością dźwięków użytą w sekwencji muzycznej;
- *N użytych dźwięków* - współczynnik zdefiniowany jako liczba różnych wysokości dźwięków użytych w sekwencji muzycznej;
- *Dźwięki w skali C-dur* - współczynnik zdefiniowany jako proporcja dźwięków ze skali C-dur do wszystkich dźwięków w sekwencji muzycznej;
- *Dźwięki w skali c-moll* - współczynnik zdefiniowany jako proporcja dźwięków ze skali c-moll do wszystkich dźwięków w sekwencji muzycznej;

Punktem odniesienia do oceny wygenerowanych przykładów muzycznych był zbiór treningowy. Przy użyciu dekodera z wytrenowanego modelu wygenerowano po 20 przykładów muzycznych dla każdej z czterech emocji (e1, e2, e3, e4), czyli po 80 przykładów dla każdego z badanych modeli. Tabela 3 przedstawia wyliczone średnie (μ) i odchylenia standardowe (σ) metryk otrzymanych dla muzyki z czterema emocjami (e1, e2, e3, e4) wygenerowanej przy użyciu bazowego modelu (CVAE-Base), zaproponowanego (CVAE-EmoReg), 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 modelu CVAE-EmoReg są tylko nieco bliższe, 9 na 16 przypadków, do metryk ze zbioru treningowego niż metryki z modelu bazowego CVAE-Base. Tylko w przypadku metryki *Dźwięki w skali C-dur* mamy poprawę metryk po zastosowaniu zaproponowanego modelu. Analizując metryki *Zakres dźwięków* i *N użytych dźwięków* widzimy jak obydwa modele generują pliki o wyższych wartościach dla emocji e1, e2, które mają wysokie pobudzenie i mniejsze dla emocji e3, e4 (niskie pobudzenie).

Obydwa modele nauczyły się korzystać z dźwięków dwóch przeciwstawnych skal dur i moll do zastosowania ich podczas generowania sekwencji muzycznych. Skale te są kojarzone 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 powiedzieć, że regulacja przestrzeni ukrytej nie wpłynęła znacząco na jakość generowanych przykładów, mimo że ułatwiła sposób wyboru emocji z przestrzeni ukrytej dla generowanej sekwencji. Jakość generowanych plików zależy bardziej od struktury sieci, liczby warstw czy jednostek rekurencyjnych [9].

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-EmoReg μ (σ)	Zbiór treningowy μ (σ)
Zakres dźwięków	e1	**8.65** (2.22)	7.75 (2.14)	9.32 (2.93)
	e2	7.85 (1.68)	**8.90** (2.84)	9.01 (2.14)
	e3	**6.30** (2.15)	6.00 (1.97)	6.19 (1.68)
	e4	6.00 (1.64)	**7.60** (3.72)	7.44 (1.95)
N użytych dźwięków	e1	**5.80** (1.03)	5.60 (1.16)	6.29 (1.36)
	e2	**5.85** (1.24)	5.55 (1.56)	6.28 (1.31)
	e3	4.15 (0.65)	**4.40** (1.07)	4.57 (0.85)
	e4	4.05 (0.67)	**4.35** (0.85)	4.84 (0.91)
Dźwięki w skali C-dur	e1	**0.96** (0.10)	0.93 (0.14)	0.97 (0.10)
	e2	0.71 (0.14)	**0.73** (0.23)	0.72 (0.13)
	e3	0.75 (0.18)	**0.77** (0.15)	0.77 (0.13)
	e4	0.99 (0.03)	**0.98** (0.04)	0.98 (0.10)
Dźwięki w skali c-moll	e1	0.69 (0.12)	**0.67** (0.14)	0.66 (0.12)
	e2	**0.91** (0.15)	0.90 (0.17)	0.91 (0.15)
	e3	0.91 (0.17)	**0.92** (0.12)	0.92 (0.15)
	e4	**0.72** (0.12)	0.73 (0.15)	0.65 (0.15)

Podsumowanie

W artykule przedstawiono nową metodę regulacji przestrzeni ukrytej wariacyjnego autoenkodera w celu powiązania tej przestrzeni z emocją generowanych przykładów muzycznych. Regulacja umożliwiła ułożenie przestrzeni ukrytej wg ćwiartek modelu emocji Russella. Ewaluacji poddano przestrzeń ukrytą, jak i generowane sekwencje otrzymane dla modelu z regulacją i bez. Model z regulacją potwierdził ułożenie przestrzeni ukrytej zgodnie z ćwiartkami modelu emocji. Ułożona wg emocji przestrzeń ukryta jest interpretowalna i ułatwia sposób wyboru emocji podczas losowego generowania nowych sekwencji.

Ewaluacja wygenerowanych przykładów muzycznych za pomocą metryk badających dźwięki i skale wykazała, że sekwencje otrzymane za pomocą zaproponowanego modelu z regulacją i modelu bez regulacji dają podobne jakościowo sekwencje muzyczne. Z powodu elementu losowego jakim jest użycie wektora wartości losowych w przestrzeni ukrytej, generowane przykłady z pomocą obydwu modeli różnią się nieco od zbioru treningowego, jednak rozłożenie wartości metryk dla poszczególnych emocji są zbliżone do danych uczących.

Zaprezentowana regulacja wariacyjnego autoenkodera może być użyta i w pozamuzycznych dziedzinach, gdzie mamy etykiety przykładów uczących i chcemy umieścić je w określonych obszarach przestrzeni ukrytej. W przyszłości można by wykonać regulację przestrzeni ukrytej w użyciem wartości emocji opisanej wartościami ciągłymi, które bardziej by odzwierciedlały odcienie emocji zawartych w generowanej muzyce. Innym zagadnieniem kontynuacji badań byłoby generowanie dłuższych sekwencji o zmiennej, kontrolowanej w czasie emocji.

Niniejsze badania zostały zrealizowane w ramach pracy nr WZ/WI-IIT/3/2023 w Politechnice Białostockiej i sfinansowane ze środków Ministerstwa Nauki i Szkolnictwa Wyższego.

***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] Ji S., Yang X., Luo J., A survey on deep learning for symbolic music generation: Representations, algorithms, evaluations, and challenges, ACM Comput. Surv., vol. 56 (2023)
[3] Kingma D. P., Welling M., Auto-encoding variational bayes, 2nd Int. Conf. Learn. Represent. (ICLR), (2014)
[4] Valenti A., Carta A., Bacciu D., Learning style-aware symbolic music representations by adversarial autoencoders, 24th Eur. Conf. Artif. Intell. (ECAI), (2020), 1563–1570
[5] Guo R., Simpson I., Magnusson T., Kiefer C., Herremans D., A variational autoencoder for music generation controlled by tonal tension, Joint Conf. AI Music Creativ. (CSMC + MuMe), (2020)
[6] Ji S., Luo J., Yang X., A comprehensive survey on deep music generation: Multi-level representations, algorithms, evaluations, and future directions, *CoRR*, (2020)
[7] 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
[8] Grekow J., Dimitrova-Grekow T., Monophonic music generation with a given emotion using conditional variational autoencoder, *IEEE Access*, vol. 9 (2021), 129088–129101
[9] Grekow J., Generowanie wielogłosowej muzyki o określonej emocji przy użyciu wariacyjnego autoenkodera, Przegląd Elektrotechniczny, 99 (2023), nr 6, 225–229
[10] Zhang D., Li X., Lu D., Tie Y., Gao Y., Qi L., Multitrack emotion-based music generation network using continuous symbolic features, IEEE Int. Conf. Multimed. Expo (ICME), (2024),1–6
[11] Russell J. A., A circumplex model of affect, *J. Pers. Soc. Psychol.*, vol. 39 (1980), no. 6, 1161–1178
[12] Grekow J., Automatyczna detekcja i wizualizacja emocji w muzyce. Rozprawa doktorska. Polsko-Japońska Wyższa Szkoła Technik Komputerowych, (2009)
[13] Hadjeres G, Nielsen F., Pachet F., Glsr-vae: Geodesic latent space regularization for variational autoencoder architectures, IEEE Symp. Ser. Comput. Intell. (SSCI), (2017), 1–7
[14] Pati A., Lerch A., Attribute-based regularization of latent spaces for variational auto-encoders, Neural Comput. Appl., vol. 33, (2021), no. 9, 4429–4444
[15] Ji S., Yang X., Muser: Musical element-based regularization for generating symbolic music with emotion, AAAI Conf. Artif. Intell., vol. 38 (2024), 12821–12829
[16] 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
[17] 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)
[18] 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
[19] Chollet F., et al., Keras, https://keras.io, (2015)