Шифр: **генератор музики**

Тема роботи: «Генерація музики за малюнком»

ЗМІСТ

ВСТУП

Сфера застосування нейронних мереж нині багатогранна і дивує своїми можливостями, наприклад, музикою. Однак у музичній генерації нейронні мережі практично не використовуються в чистому вигляді, частіше як проміжна або допоміжна ланка. Але варто відзначити, що на теперішній час все ж таки існують досить успішні розробки, що генерують музику в стилі відомих композиторів, джазові варіації, електронну музику та багато іншого.

Дослідження включатиме аналіз архітектури та параметрів LSTM мережі для досягнення оптимальних результатів з генерації музики.

Крім того, робота також пропонує використання LSTM мережі у поєднанні з іншими методами машинного навчання для поліпшення якості музичної композиції

Результати цього дослідження можуть бути корисними для музичних композиторів, аранжувальників та інших професіоналів у галузі музики, які прагнуть до використання передових методів машинного навчання для створення оригінальної та виразної музики.

РОЗДІЛ 1
ЗАГАЛЬНОСИСТЕМНІ ПИТАННЯ. ПОСТАНОВКА ЗАДАЧІ НА ПРОЕКТУВАННЯ

1.1 Огляд і аналіз принципів створення музичної композиції

Незважаючи на всі досягнення у розумінні творчих процесів, створення музики не може проходити автоматично. Роль користувача-композитора дуже висока, і можна лише говорити про автоматизацію цього процесу. Передається музикою та картинами емоційність складно розпізнається [1]. Сам процес створення музики на даний момент не піддається формалізації, хоч і ґрунтується на чітко визначених музичних правилах. Здебільш цей процес пов'язано з вивченням і повторенням різних музичних стилів.

Для ознайомлення з музичним матеріалом використовують візуальне відображення його в класичній нотній нотації. Будь-яку музику, зафіксовану на папері за допомогою нотного стану, можна вважати графіком зміни висоти звуків з часом. Нотний стан можна сприяти як систему координат, на горизонтальній осі якої позначається час, на вертикальної – висота нот (рис. 1.1).

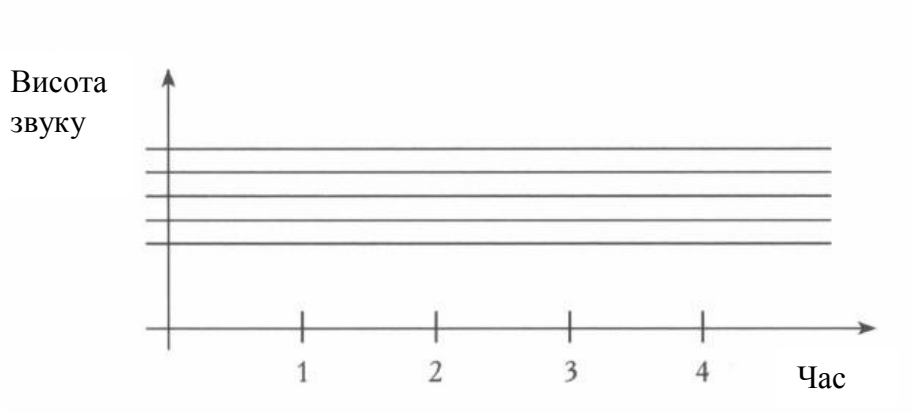


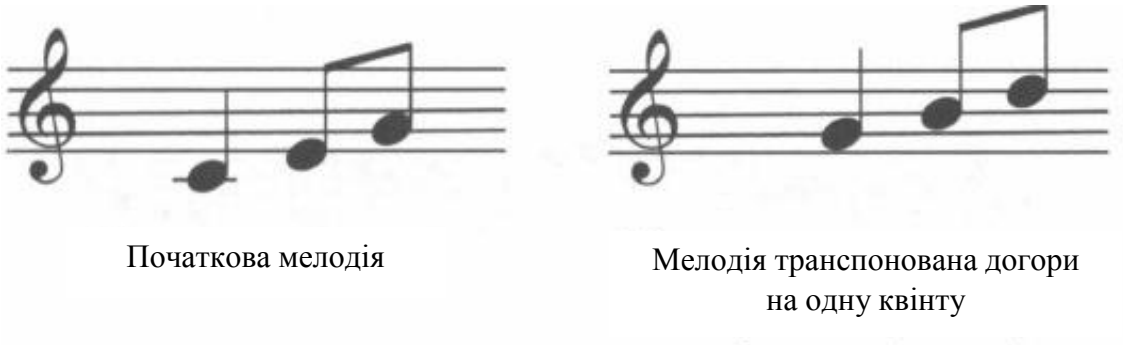
Рисунок 1.1 – Нотний стан

Тривалість звуків позначається за допомогою нот. Висота звуків позначається положенням головки ноти на лінії нотного стану або у проміжку між лініями. А для знання абсолютної висоти звуків потрібний ключ.

Існує три різні ізометричні перетворення на площині: перенесення, відображення та поворот [2].

Переносом будемо вважати геометричне перетворення, при якому всі точки фігури переміщуються в заданому напрямку на ту саму відстань, при цьому форма фігури не змінюється.

Ізометричне перенесення нот уздовж вертикальної осі називається транспозицією. У результаті транспозиції виходить та сама мелодія, але більш висока або нижча, залежно від напрямку перенесення (рис. 1.2).



Початкова мелодія

Мелодія транспонована догори на одну квінту

Рисунок 1.2 – Ізометричне перенесення нот уздовж вертикальної осі

Відображення – це перетворення, яке замінює фігуру її дзеркальним відбитком. Комбінація відображень щодо горизонтальної та вертикальної осі є поворотом на 180°. При відображенні щодо горизонтальної осі виходить інверсійна мелодія.

Також існує ізометричне перетворення, яке застосовується в музиці – масштабування. Найбільш наочними прикладами цього перетворення є стиск та розтягнення вздовж тимчасової осі. Вертикальне масштабування досить складно виконати і почути у музичній композиції. При вертикальному масштабуванні всі інтервали пропорційно розширюються.

1.2 Аналіз ступеня вивченості теми

Розглянемо їх історичний розвиток. Застосування Марківських процесів у генерації музичних конструкцій вперше було розглянуто Гаррі Ф. Олсеном у 1950 році [3]. Марківські моделі через свою структуру надають тільки опис контексту залежностей через можливість переходу символів у прямій послідовності, це означає, що вони краще підходять для моделювання одновимірних символьних послідовностей.

Породжуюча граматика - формалізм генеративної лінгвістики, пов'язаний з вивченням синтаксису [4]. У рамках цього підходу формується система правил, за допомогою яких можна визначити, яка комбінація слів оформляє граматично правильну пропозицію.

Системи Ліденмайєра Л-системи лежать на стику таких сфер математики, як еволюційні методи та формальна граматика. В основі роботи Л-систем лежить набір правил заміщення, що рекурсивно застосовується на початковий рядок символів і інтерпретує кінцевий рядок як структурні елементи організму. Застосування Л-систем для генерації алгоритмічних композицій передбачає використання замість символів певних музичних параметрів.

Генетичний алгоритм Генетичний алгоритм (ГА) – це евристичний алгоритм пошуку, який використовується для вирішення завдань оптимізації та моделювання шляхом випадкового підбору, комбінування та варіації шуканих параметрів з використанням механізмів, що нагадують біологічну еволюцію [5]. Доцільність застосування ГА для моделювання музичної творчості обґрунтували професор кафедри комп'ютерних наук Гонг-Конгського університету Ендрю Хорнер та професор кафедри індустріальної інженерії Університету Іллінойсу Девід Голдберг у 1991 році [6].

Штучні нейронні мережі (ШНМ) - математичні моделі, а також їх програмні або апаратні реалізації, побудовані за принципом організації та функціону-

вання біологічних нейронних мереж. Перший приклад такої мережі використовував тришарову штучну рекурсивну нейронну мережу, призначену для створення тимчасової послідовності виходів, що кодують монофонічну мелодію.

Під системами, заснованих на знаннях, розуміються різні системи, засновані на правилах, які використовують уявлення знань як більш-менш структурованих символів.

1.3 Методи машинного навчання для генерації музики

Існує кілька методів машинного навчання, які використовуються для генерації музики. Вони включають:

а) глибокі нейронні мережі (Deep Neural Networks, DNN) які моделюють складні залежності між нотами, акордами та ритмами [7]. Такі мережі можуть бути навчені на великому наборі музичних даних та використовуватись для генерації нових мелодій, гармоній та ритмів;

б) LSTM мережі (Long Short-Term Memory Networks) є підтипом глибоких нейронних мереж, які особливо добре підходять для моделювання послідовних даних, таких як музика [8]. Вони здатні зберігати довготривалі залежності в музичних послідовностях і генерувати нові музичні фрази, мелодії та ритми;

в) генетичні алгоритми (Genetic Algorithms) які використовують для еволюційного генерування музики [9]. Вони працюють шляхом створення популяції музичних "генотипів" та поступової модифікації їх шляхом схрещування та мутації. Кращі композиції відбираються на основі заданої функції оцінки, що дозволяє отримати нові, унікальні музичні твори;

г) варіаційні автоенкодери (Variational Autoencoders, VAE) є моделями глибокого навчання, які можуть використовуватися для генерації музики шляхом управління внутрішнім простором представлень [10]. Вони можуть здійснювати варіацію музичних патернів та стилів, дозволяючи створювати нові твори;

д) трансформери (Transformers) вони демонструють великий потенціал у генерації музики і здатні моделювати довготривалі залежності і керувати

контекстом при генерації музичних послідовностей, що дозволяє їм створювати складні та виразні композиції [11].

1.4 Аналоги

Що стосується практичних реалізацій методів машинного навчання в сфері генерації музики, можна привести кілька прикладів.

Нейромережу Mubert можна використовувати через браузер або у вигляді програми [12]. Вона безкоштовно генерує музику на основі уподобань користувача. Під час створення нової композиції необхідно англійською мовою ввести характеристики в полі Enter promt. Можна прописати інструменти, жанр та встановити тривалість композиції у рядку Set duration.

Soundraw ще одна нейромережа-композитор, яка надає безкоштовну можливість складати музику [13]. Для створення треку потрібно зайти на сайт Soundraw і вибрати параметри майбутньої композиції.

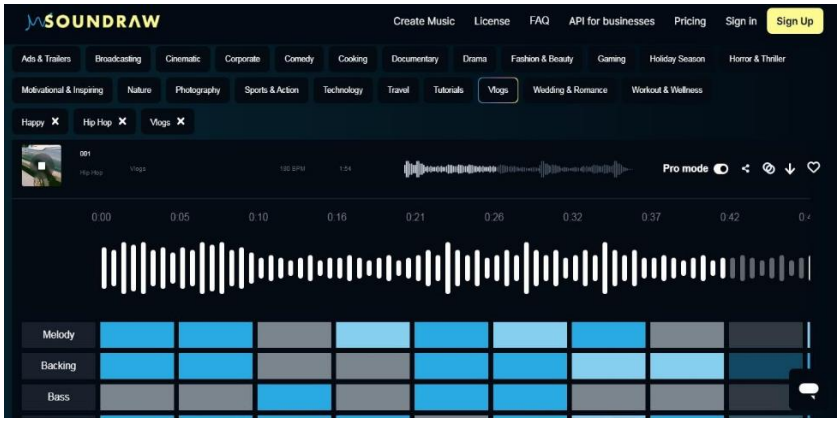


Рисунок 1.3 – Нейромережа Soundraw для генерації музики

Заздалегідь необхідно визначитися з настроєм бажаної музики (Mood): похмуре, романтичне, умиротворене, щасливе представлено понад 20 варіантів. Потім вибрати жанр композиції (Genre): хіп-хоп, рок, фанк, хаус тощо.

Jukebox ще одна програма від компанії OpenAI, що стала особливо відомою після запуску ChatGPT.

Для створення музики в ній потрібні знання в галузі програмування, досить багато часу та високі обчислювальні потужності – простіше кажучи, продуктивний комп'ютер. Мережа поки що не підходить для тих, хто хоче швидко і без особливих навичок створити трек для відео.

Нейромережа AIVA здатна створювати музичні композиції, що її вже називають конкурентом для композиторів-людей [14].

Неромережа Melobytes допоможе написати пісню, створити музику по картинці та багато іншого [15].

Мережа буде цікава тим, хто починає вивчати можливості ШІ та нейромереж. Вона пропонує велику кількість налаштувань, при цьому експериментувати та зберігати створену музику можна безкоштовно.

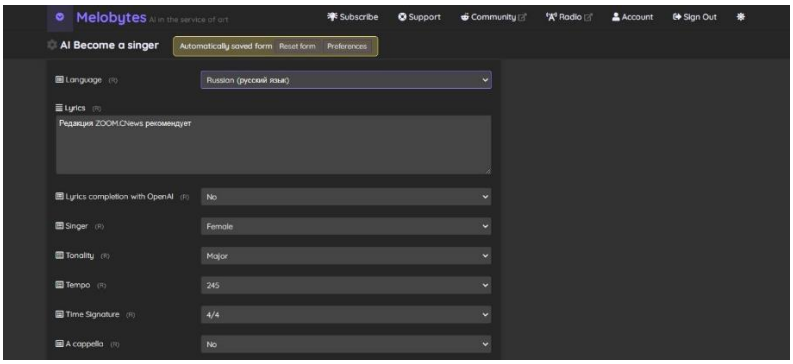


Рисунок 1.4 – Нейромережа Melobytes для генерації музики

1.4 Постановка задачі. Технічне завдання на розробку

Основною задачею роботи є розробка системи, що матиме функціональний модуль для генерації музики за малюнком з використанням методів машинного навчання. Для забезпечення функціоналу буде розроблено LSTM мережу. Для виконання поставленої задачі необхідно вирішити наступні задачі:

- проаналізувати програми для створення звукових послідовностей;
- знайти та вивчити методи та підходи автоматизованої генерації звукових послідовностей по аналізу зображень;
- розробити модуль генерації звукових послідовностей та синтезу звуків.

Висновки за розділом

У першому розділі було наведене огляд існуючих підходів до вирішення проблеми створення звукового ряду по зображенню. Був здійснений огляд наукових джерел, та розглянуті підходи до вирішення цієї проблеми. Було виконано огляд існуючих додатків для створення звукового ряду по зображенню. За результатами роботи подібних систем було сформулювати основні вимоги до LSTM нейронної мережі на основі якої буде побудовано модуль для створення звукового ряду по зображенню.

РОЗДІЛ 2
ПРОЕКТНІ І ТЕХНІЧНІ РІШЕННЯ. ВИДИ ЗАБЕЗПЕЧЕННЯ

2.1 Математична модель музичної форми

Форма кожного музичного твору завжди індивідуальна, властива саме йому та неповторна. Але в той же час кількість законів і правил освіти форми відносно обмежена, і завдяки цьому безліч творів має загальні ознаки в будові. Це дає можливість вивести типи форм або загальні композиційні схеми, що набули значного поширення внаслідок своєї гнучкості та доцільності, Основне значення для освіти музичної форми мають мелодія, гармонія та ритм у їх взаємодії. У сучасній професійній музиці організація звукової мови здійснюється шляхом з'єднання первинних мотивів у фрази, речення, періоди та цілі розділи [16]. Проста трьох частинна форма складається шляхом зіставлення трьох періодів, у тому числі третій є повторення першого (репризу). Складні форми утворюються: а) шляхом зіставлення простих форм (включаючи період); б) шляхом використання у них підсобних розділів – зв'язок, вступів, висновків, розробок.

На рис. 2.1 зображено схему музичного твору з використанням введених раніше понять.

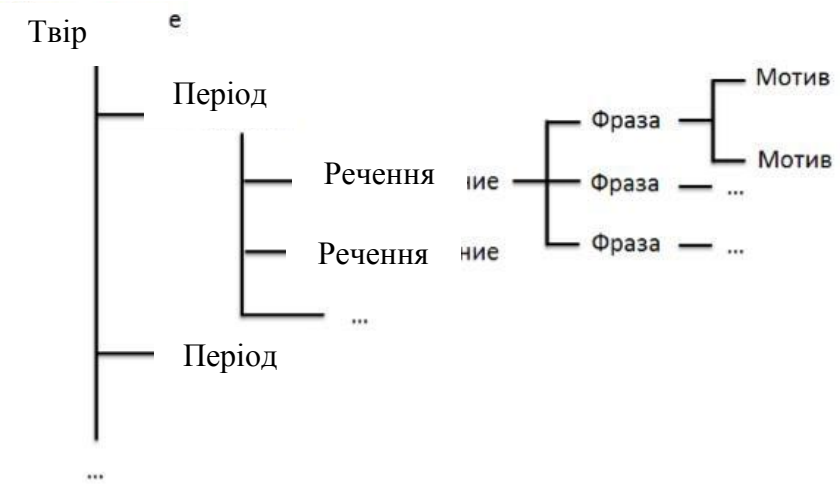


Рисунок 2.1 – Схема музичного твору

Патерн - закономірна регулярність, шаблон, зразок, що повторюється. Музичний патерн – не прив'язана до будь-якої тональності або акорду послідовність двох або більше нот (або дрібніших патернів), що мають відносну тривалість.

Найпростішими патернами є дві ноти, пов'язані будь-яким із чотирьох вищезгаданих правил для зв'язку двох нот. Наведемо найпростіші патерни, що найчастіше використовуються (рис. 2.11).

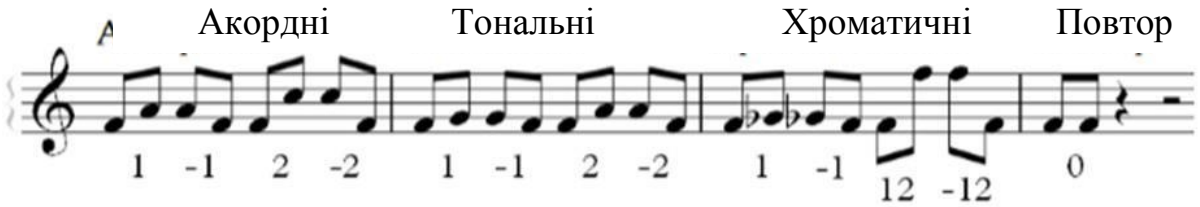


Рисунок 2.2 – Простий патерн

Складовими патернами називають комбінацію з двох і більше патернів (як найпростіших, так і складових). На рис. 2.3 представлені варіанти різних складових патернів.

З двох простих патернів

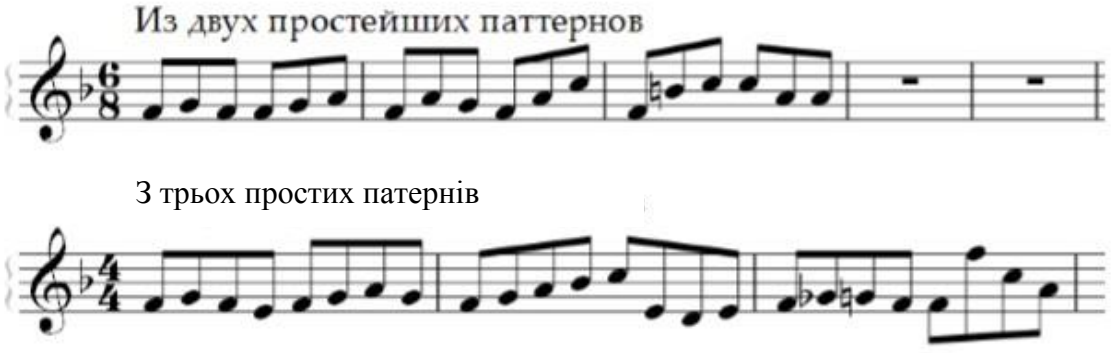


Рисунок 2.3 – Складові патерну

Процес побудови мотиву відбуватиметься у три етапи.

Етап перший:

– визначається загальна тривалість мотиву (у метричних одиницях);

– вибирається ритмічна структура мотиву (тривалості опорних нот в метричних одиницях);

– вибирається перша нота, з якої буде починатися мотив;

– покроково накладаються найпростіші патерни (вже в потрібних метричних одиницях), вибираючи кожен наступний патерн довільним чином, але з чергуванням руху вгору і вниз, щоб зрештою не виходило тільки рухів догори або донизу;

– у результаті виходить мотив із нот рівних тривалостей (рис. 2.4).

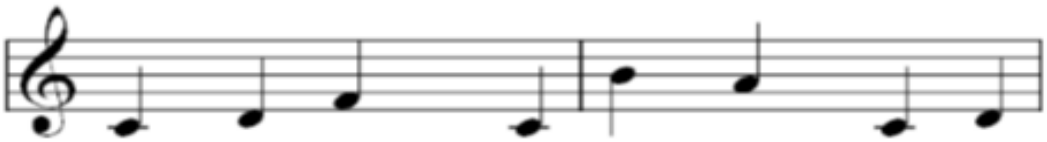


Рисунок 2.4 – Мотив із нот рівних тривалостей

Етап другий:

– на цьому проході деякі з нот замінюються найпростішими та складовими патернами, що обираються на основі відстані між сусідніми нотами. В результаті

виходить мотив, що складається з нот різних тривалостей і вже претендує на звання завершеного мотиву (рис. 2.5).

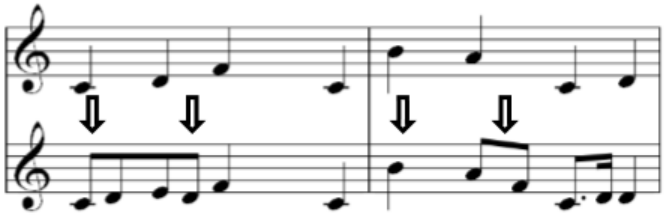


Рисунок 2.5 – Завершений мотив з нот різних тривалостей

Етап третій;

– починається третій прохід за мотивом: аналізуються зв'язки між нотами та проводиться заміна деяких найпростіших патернів на еквівалентні складові; додаються мелізми (форшлаги та трелі) до деяких нот, що задовольняють умови мелізмів;

– у результаті виходить завершений мотив, який може бути основою для подальшої побудови мелодії та музичного твору в цілому (рис. 2.6).

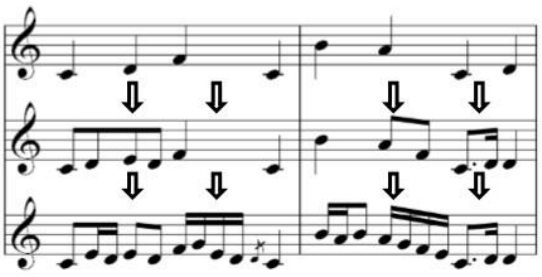


Рисунок 2.6 – Побудови мелодії та музичного твору в цілому

Тепер, коли є готовий мотив, можна рухатися далі та складати музичну фразу.

Твір будується з періодів (кожен із них включає дві пропозиції) за схемою A-B-A', тож другий період будується незалежно від першого, а третій є

модифікацією першого. У загальному вигляді алгоритм генерації другого мотиву у творі зображено рис. 2.7.

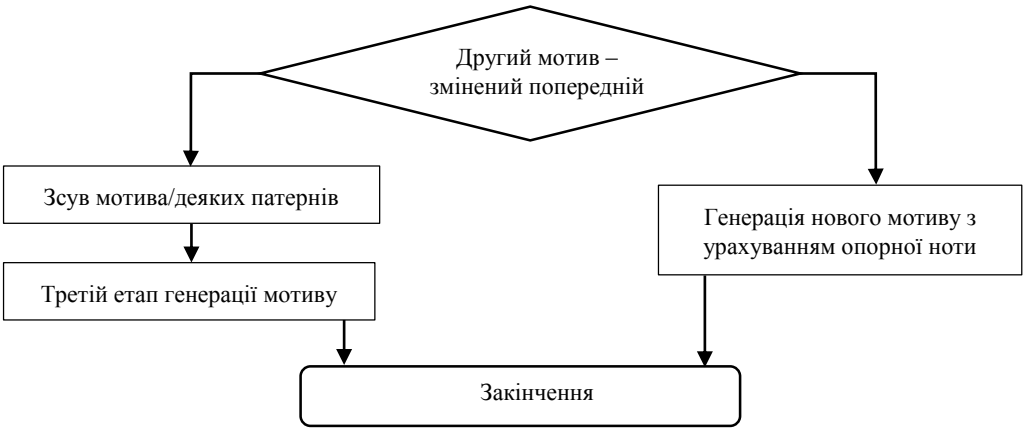


Рисунок 2.7 – Алгоритм генерації другого мотиву у творі

2.2 Алгоритм гармонізації мелодії

У класичній музиці під терміном «гармонізація мелодії» розуміється пошук акордової сітки та побудова голосів до повного чотириголосства. У рамках нашої моделі під «гармонізацією мелодії» розумітимемо лише підбір акордового супроводу.

Підбір здійснюється шляхом пошуку серед усіх акордів щаблів поточної тональності тих, які включають поточний звук мелодії. Так, нота «мі» в тональності до-мажор входить до складу акордів I, III та VI ступенів. Інші приклади входження нот до складу акордів показано на рис. 2.8.

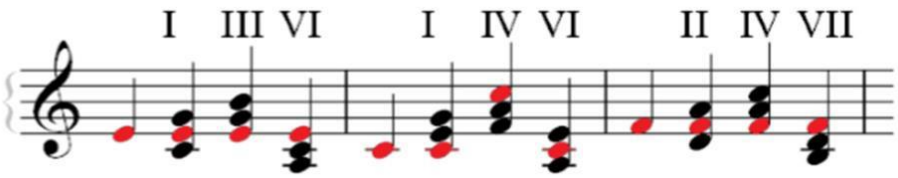


Рисунок 2.8 – Приклад входження нот до складу акордів

Кожна нота в гамі входить до складу трьох різних тризвучій (як перша, друга та третя ноти тризвучтя відповідно). Схема допустимості застосування одних акордів за іншими представлена рис. 2.9.

I → II, III, IV, V, VI, VII
II → I, III, V
III → I, IV, V, VI, VII
IV → I, II, V
V → I, III, VI, VII
VI → I, II, III, IV
VII → I

Рисунок 2.9 – Схема допустимості застосування одних акордів за іншими

Виходячи з списку допустимих акордових послідувань, складемо алгоритм гармонізації мелодії:

а) мелодія розбивається на такти. Робота з кожним тактом ведеться окремо;

б) усередині кожного береться нота мелодії. По ній вибирається список акордів, до складу яких вона входить, а потім цей список звіряється зі списком "послідовників" попереднього акорду (якщо такий є);

в) процес повторюється кожної ноти. Щойно остання нота останнього такту гармонізована, процес гармонізації зупиняється.

2.3 Загальний опис модуля для генерації музичних фрагментів

Модуль призначений для генерації музичних фрагментів без втручання користувача, застосовуючи різні методи штучного інтелекту в контексті алгоритмічної композиції. Структура включає в себе п’ять модулів (рис. 2.10), які взаємодіють між собою.

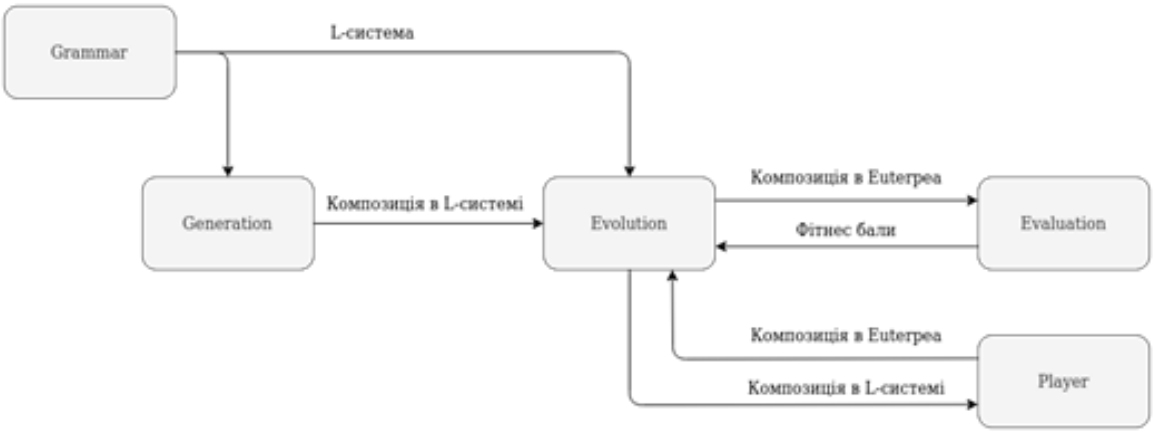


Рисунок 2.10 – Структура модуля для генерації музичних фрагментів

Модуль Grammar описує формальну граматику (L-систему) для створення композицій та надає зручний спосіб представлення музичних фрагментів і запитів користувача, які використовуються в інших частинах системи.

Модуль Generation відповідає за породження випадкових композицій за даними запиту в межах визначеної граматики, які потім вдосконалюються.

Модуль Evolution реалізовує еволюційний алгоритм, який випадковим чином ініціалізує початкову популяцію композицій та ітеративно застосовує оператори мутації та схрещування до них.

Створені композиції модуль Player експортує у музичні фрагменти бібліотеки Euterpea та має можливість збереження фрагмента у MIDI-файл.

Функцію оцінки композицій надає модуль Evaluation - система на основі правил, яка приймає композиції та повертає відповідну суму балів.

Таким чином, граматика обмежує простір композицій до такого, в якому не допускаються грубі порушення гармонії. В той час як еволюційний процес шукає такі композиції, для яких система правил дає якомога більше балів, тобто які характеризуються більшою благозвучністю в межах заданого простору.

2.4 Розробка архітектури нейронної мережі

У створюваній мережі використовується чотири різні типи шарів, рис. 2.11.

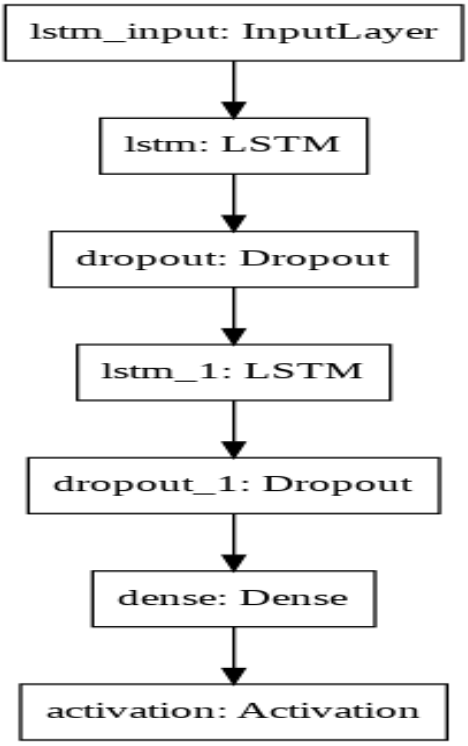


Рисунок 2.11 – Схема моделі LSTM-неейронної мережі

За функцію активації взято softmax (рис. 2.12).

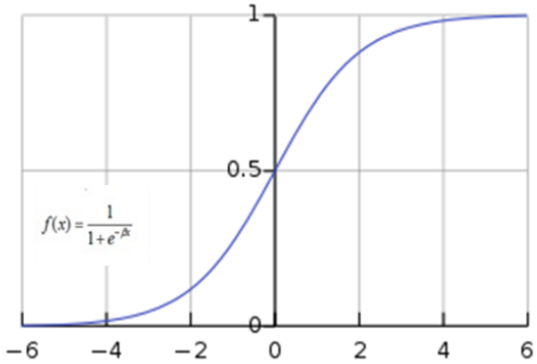


Рисунок 2.12 – Функція softmax

Усі параметри навчання наведено у табл. 2.1.

Таблиця 2.1 — Параметри навчання нейронної мережі

Параметр	Значення
Activation	softmax
Loss	categorical_crossentropy

Optimizer	rmsprop
Metrics	accuracy
Batch size	128
Epochs	200

Детальна схема топології розробленої нейронної мережі наведена на рис. 2.13.

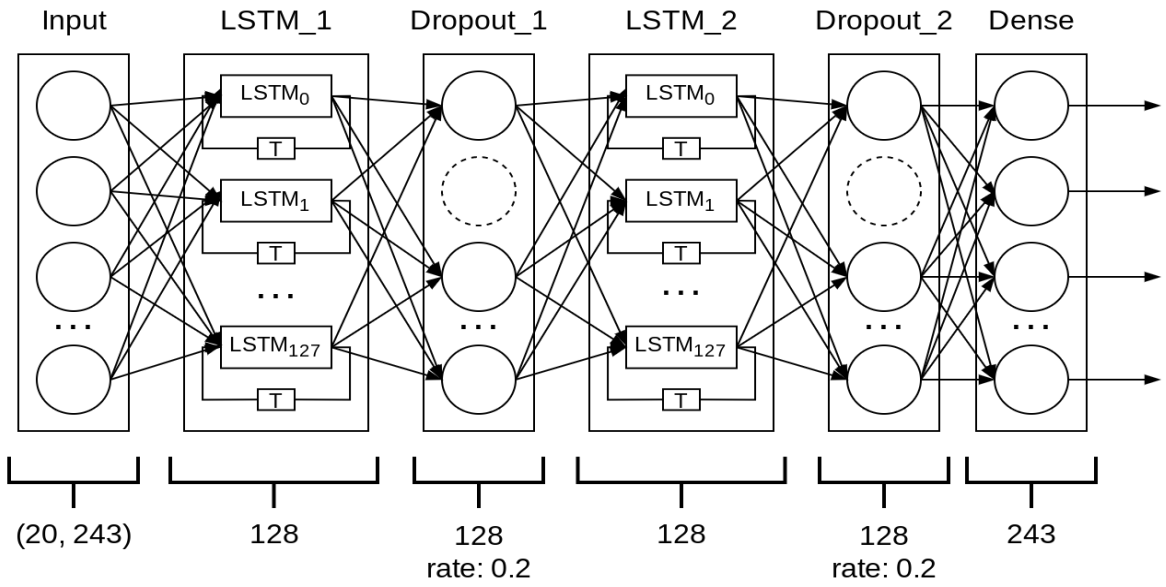


Рисунок 2.13 – Схема топології розробленої нейронної мережі

Створена нейронна мережа майже вдвічі скорочує кількість параметрів з попередньої операції. Це обумовлене тим, що нейронна мережа не розгалужується на дві осі — вісь часу та вісь нот. Час навчання мережі за рахунок цього також значно зменшується. При цьому точність (accuracy) передбачених результатів залишається на тому ж рівні.

Таким чином, вважаємо, що задача по проектуванню ефективної архітектури нейронної мережі була вирішена в повному обсязі.

РОЗДІЛ 3
ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1. Огляд обраних інструментів розробки

Для успішної реалізації модулю, необхідно проаналізувати та обрати доступні технічні засоби. До ключових технологій необхідних для розробки проекту можна віднести: мови програмування, фреймворки та бібліотеки для реалізації проекту як нейронної мережі та її навчання.

Для проектування та навчання нейронної мережі було обрано мову програмування Python.

Мова Python поставляється з багатьма бібліотеками та фреймворками, які полегшують кодування. Це також економить значну кількість часу.

Найпопулярнішими бібліотеками є:

- NumPy, який використовується для наукових обчислень;
- SciPy для більш складних обчислень;
- Scikit, для вивчення інтелекту та аналізу даних [17].

Ці бібліотеки працюють разом із потужними фреймворками, такими як:

- TensorFlow;
- CNTK;
- Apache Spark.

Ці бібліотеки та фреймворки необхідні, коли мова йде про проекти машинного та глибокого навчання.

Крім того, Python є мовою програмування з відкритим кодом і користується підтримкою з багатьох ресурсів має якісну документацію по всьому світу. Він також має велику та активну спільноту розробників, які надають свою допомогу на будь-якому етапі розвитку.

3.1.1 Середовище програмування

В якості середовища для навчання та тестування нейронної мережі було обрано інструмент Google Colab [18].

Colab — це безкоштовне середовище для Jupyter Notebook, яке повністю працює у хмарі. Найголовніша перевага цього середовища в тому, що воно надає можливість безкоштовно користуватися ресурсами для обробки Big Data, дозволяючи навчати нейронні мережі на потужних GPU.

На даний момент існують два основні фреймворк, які використовуються в Python: Django та Flask.

В результаті порівняння фреймворків Django та Flask – було вирішено використовувати для розробки Django, оскільки він більше підходить для даного проекту. Flask було відхилено через малу кількість вбудованих можливостей, та через це – повільнішу швидкість розробки. Також перевагою Django є вбудовані механізми забезпечення безпеки та зручна структура організації коду, яка дозволяє дуже швидко масштабувати проект та додавати нові функції.

3.2 Програмна реалізація модуля

Ми використовуємо вхідні дані у форматі ABC, як було розглянуто у 2 розділі.

Зразкові рядки наведено на рис. 3.1

```
[V: S] (BA) !p!G2 |z AGA|(FG) A2|
w: ple-na, Do-mi-nus te-cum,
[V: A] F2 E2|z FEC|(DE) F2 |
w: ple-na, Do-mi-nus te-cum,
[V: T] (dc) c2|z ccA|(Ac) c2 |
w: ple-na, Do-mi-nus te-cum,
[V: B] (B,,F,) C,2|z F,C,F,|(D,C,) F,2 |
w: ple-na, Do-mi-nus te-cum,
```

Рисунок 3.1 – Зразкові рядки мелодії

Масив даних для навчання мережі було обрано Making Music with ABC 2 [19].

Зчитування файлу виконується за допомогою наступного коду в Google Colab.

```
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
with open('my_song.abc', 'r') as f:
    text = f.read()
```

Для прослуховування пісні встановлюємо додаткові інструменти у середу.

```
!apt-get install -y -qq abcmidi timidity
```

Для збереження обраної пісні використаємо наступний код:

```
song = text.split('\n\n')
with open('my_song.abc', "w") as f:
    f.write(‘song’)
```

Спочатку ми конвертуємо її в файл mid, а потіму формат wav.

```
!abc2midi "my_song.abc" -o "my_song.mid" && timidity "my_song.mid" -Ow "my_song.wav"
```

Далі для генерації музики побудуємо LSTM нейронну мережу за структурою наведеною в 2 розділі.

Спочатку створимо навчальну вибірку

```
#Унікальні символи, знайдені в піснях.
vocab = set(text)
```

```
# словник: ключ = символ, значення = індекс, вказавши символ, ми отримуємо його індекс
char_to_index = {char_:ind for ind, char_ in enumerate (vocab)}
ind_to_char = np.array(vocab)
text_as_int = np.array([char_to_index[c] for c in text])
#'X:1\nT:dfkjds' ----- > [49 22 13 0 45 22 26 67 60 79 56 69 59]
```

Далі для генерації послідовності створюємо навчальні послідовності

input: рядок із 100 символів

target: рядок зі 100 символів, але зрушений на 1.

Модель буде доручено навчитися прогнозувати наступний знак на основі 100 попередніх. Це буде модель RNN версії "many to many", яка насправді прогнозуватиме один наступний символ, але в процесі навчання помилка буде враховуватися по всій послідовності (100 передбачень).

```
seq_length = 100
step = 10
sequences = np.array([text_as_int[i:i+seq_length+1] for i in range(0, len(text_as_int)-seq_length-1,step)])
input_text = np.array([seq[:-1] for seq in sequences])
target_text = np.array([seq[1:] for seq in sequences])
```

LSTM мережа:

```
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Input, LSTM, Dense, Embedding
vocab_size = len (vocab)
#new value
embedding_dim = 256 * 2
rnn_units = 1024*2
x = Input(shape=(seq_length,))
```

e = Embedding (vocab_size, embedding_dim) (x)

l = LSTM(rnn_units, return_sequences=True)(e)

d = Dense (vocab_size, activation = 'softmax') (l)

model = Model(inputs=x, outputs=d)

model.summary()

Отриманий результат, структуру моделі, представлено на рис. 3.2

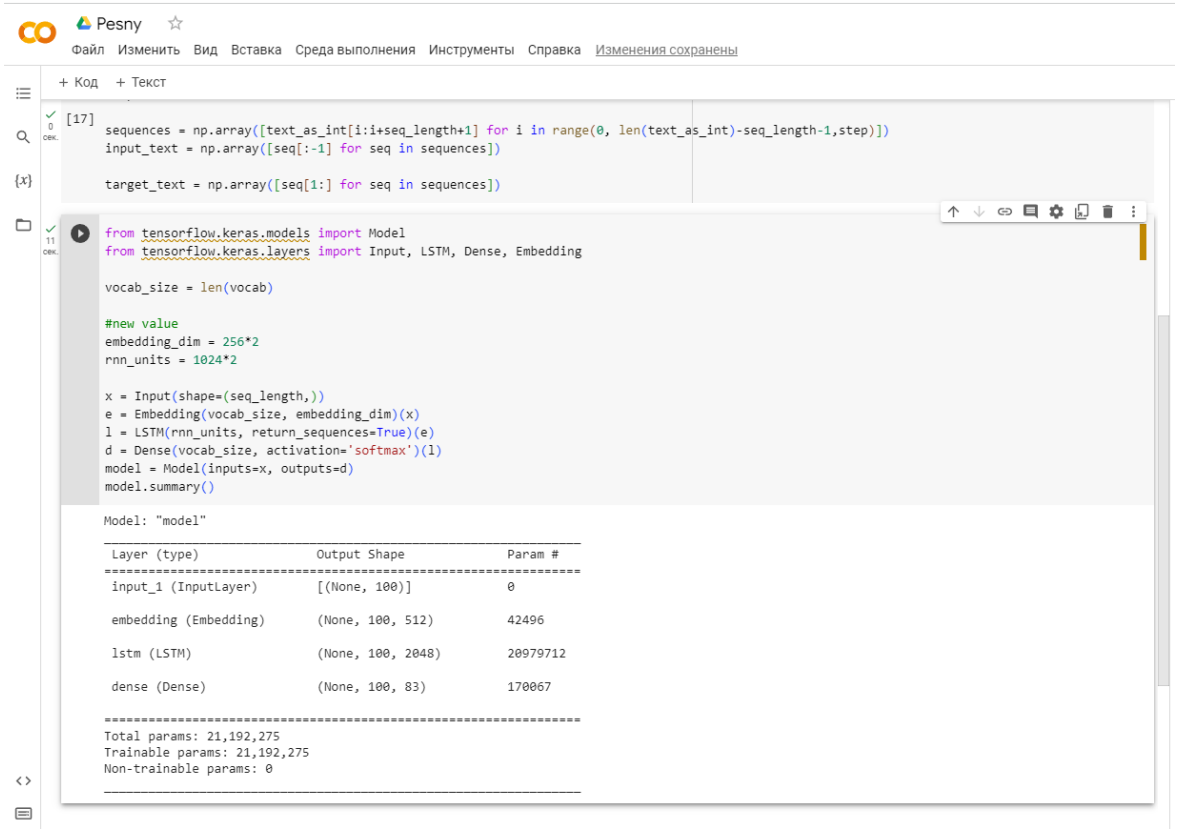


Рисунок 3.2 – Структура побудованої LSTM моделі

Далі потрібно провести навчання мережі:

from tensorflow.keras.optimizers import Adam

model.compile(optimizer=Adam(), loss='sparse_categorical_crossentropy')

EP=5

BS = 128

```
hist = model.fit(input_text, target_text, batch_size=BS, epochs=EP)
```

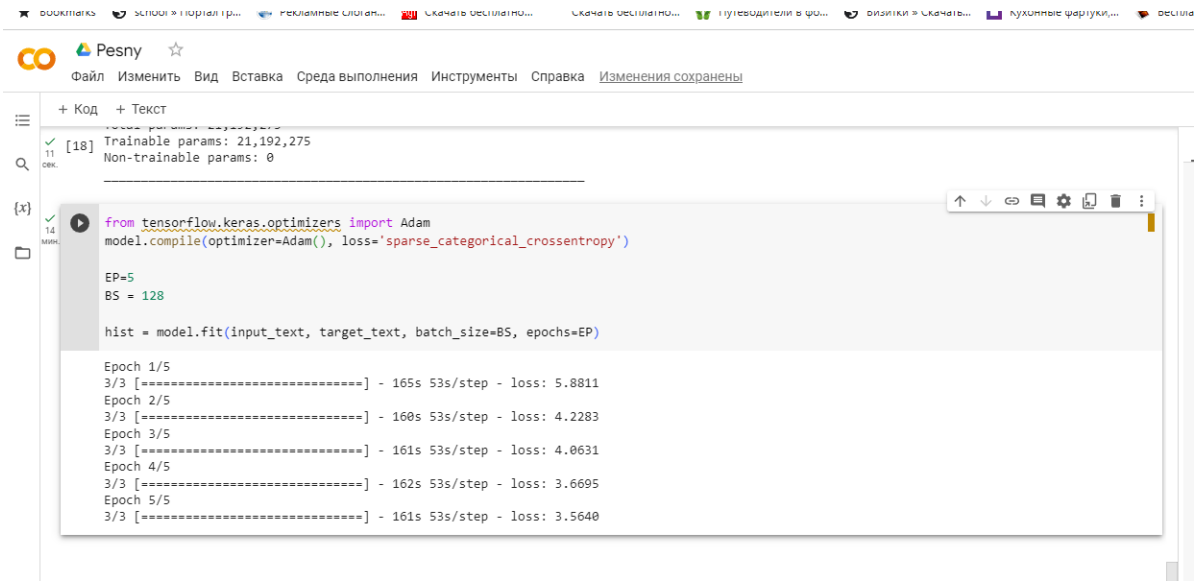


Рисунок 3.3 – Результат навчання моделі

Створення музики використовуємо вже навчану модель:

```
def generate_text(model, start_string, generation_length=100):
  input_eval = np.array([char_to_index[s] for s in start_string])
  x = np.zeros((1, seq_length))
  x[0,-len(input_eval):] = input_eval[:]
  text_generated = []

  model.reset_states()
  for i in range(generation_length):
     predictions = model.predict(x)[0,-1]
     predictions = predictions.astype(np.float64)
     predictions = predictions/np.sum(predictions)
     predicted_id = np.argmax(np.random.multinomial(1, predictions))
     x[0,:-1] = x[0,1:]
     x[0,-1] = predicted_id
```

```
            text_generated.append([ind_to_char(predicted_id)])
        return (start_string + ''.join(text_generated))
        new_song = generate_text(model, "X:", generation_length=500)
```

Щоб побачити результат потрібно виконати команду:

```
        with open('new_song.abc', "w") as f:
            f.write(‘new_song’)
        !abc2midi "new_song.abc" -o "new_song.mid" && timidity "new_song.mid" -
Ow "new_song.wav"
        Audio('new_song.wav')
```

3.3 Навчання нейронної мережі

Навчання моделі нейронної мережі потребує багато обчислювальних ресурсів, та велику кількість якісних навчальних даних, але чим більше даних буде використовуватися, тим більше часу та обчислювальної потужності необхідно моделі для завершення навчання [20].

Для прискорення навчання були використані хмарні обчислення та робились за допомогою ресурсів графічного процесора з використанням можливостей Google Colab. Це видно з результатів порівняння зображених на рисунку 3.4. Для порівняння були використані параметри за замовчуванням, та додатковий графічний процесор з 2 гігабайтами оперативної пам'яті.

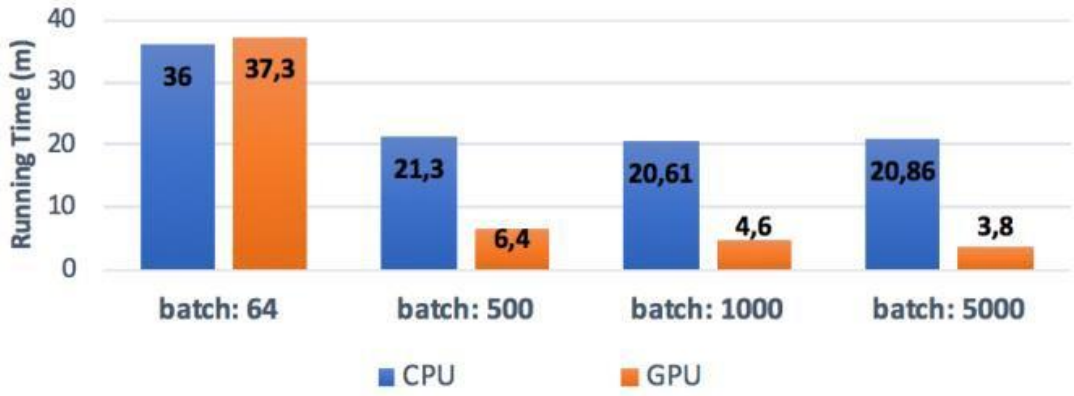


Рисунок 3.4 – Порівняння швидкості навчання нейронної мережі з використанням CPU та GPU

Як видно з графіків, з зростанням кількості даних – зростала швидкість їх обробки, особливо використовуючи графічний процесор, який при максимальній кількості даних був швидше ніж центральний процесор в 5,5 разів.

Вибір даних для тренування є важливою складовою будь-якої нейронної мережі, оскільки вона отримує всі свої «знання» про музику лише з навчальних даних. Тому чим більше буде даних для навчання, та чим якісніше вони будуть, тим кращими будуть результати, згенеровані нейронною мережею. Також необхідно мати дані для тестування нейронної мережі, які повинні відрізнятись від даних для тренування, щоб визначати, чи змогла модель абстрагуватись від даних тренування та скласти нові мелодії, а не просто відтворювати дані, якими її навчали.

До тренувальних даних є дві вимоги. Перша – музичні композиції повинні бути в одному жанрі та написані людьми, а не іншою нейронною мережею. Друга вимога – мелодії повинні приємно звучати. Оскільки зазвичай різні жанри музики для людей звучать по-різному та залежать від їх смаку, було вирішено обрати для навчання класичну музику як найбільш нейтральний жанр. До того ж класична музика – це найдоступніший музичний жанр у форматі MIDI.

Для збору навчального дата сету було написано декілька скриптів, за допомогою яких були зібрані файли з найбільшого сайту агрегатора класичної

музики – «Classical Piano Midi Page», де представлені композиції таких композиторів як Моцарт, Бах, Бетховен та багато інших.

Для навчання моделі було необхідно було перетворити зібрані MIDI-файли у формат, який зможе зрозуміти нейронна мережа. Файли MIDI надають інформацію про час початку гри та зупинки на кожну ноту, а також інформацію про музичний інструмент, значення гучності та темпу гри. Але оскільки цей формат має специфічні позначення, які не дуже зручно прогнозувати, було написано функцію, яка може перекодувати їх до більш зручного текстового формату, який може зрозуміти модель нейронної мережі. Також на виході нейронної мережі музичні композиції подаються спочатку у вигляді власного текстового формату, а далі кодуються в формати MIDI та MP3.

3.4 Результати роботи моделі

Під час тестування роботи нейронної мережі, було досліджено вплив зміни деяких параметрів нейронної мережі, на функцію втрати.

Експериментальним шляхом було визначено, що оптимальними характеристиками рекурентної нейронної мережі для генерації музики є мережа, яка складається з 4 LSTM шарів, кожен розмірністю по 600 нейронів, зі значенням показника виключення (dropout) = 1 та кількістю епох 150 [21].

Фінальний графік функції втрати нейронної мережі з обраними характеристиками, зображено на рисунку 3.5.

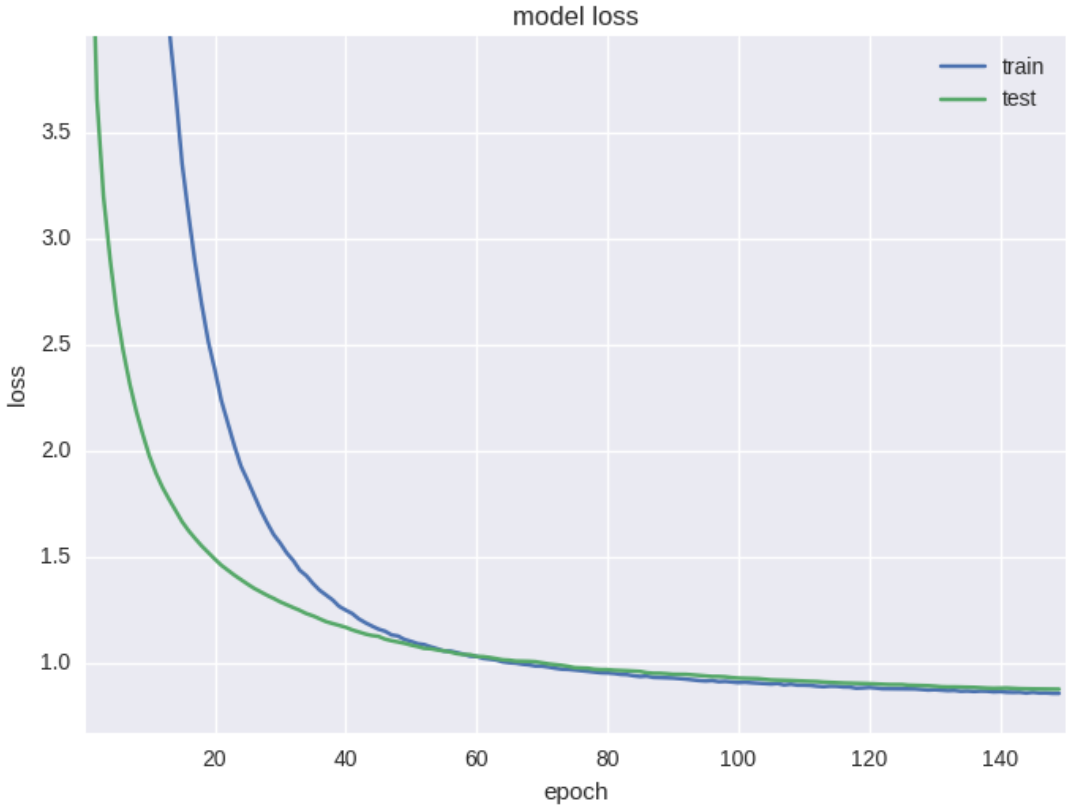


Рисунок 3.5 – Графік функції втрати для фінальної моделі нейронної мережі

ВИСНОВКИ

Використання штучного інтелекту та нейромереж дозволяє створювати музику що розширює креативні можливості композиторів і артистів.

Ця галузь все ще розвивається, і існують технічні та творчі виклики в генерації музики. Також важливо враховувати, що такі технології можуть мати певні обмеження і вимагати додаткового оброблення або налаштування, щоб досягти бажаного звучання.

У роботі розглянуто процес створення модуля для генерації музики з використанням рекурентних LSTM нейронних мереж. Для успішної реалізації цієї задачі був проведений аналіз існуючих рішень, а також визначені переваги та недоліки існуючих можливостей реалізації. У результаті розроблено модуль для генерації музичних композицій, який в подальшому може використовуватись для створення нових видів музики або в інших креативних цілях.

Результати цього дослідження можуть бути корисними для музичних композиторів, аранжувальників та інших професіоналів у галузі музики, які прагнуть до використання передових методів машинного навчання для створення оригінальної та виразної музики

Також планується подальша оптимізація налаштування параметрів моделі та навчання LSTM меріжі.

Поставленні питання дослідження вирішені в повному обсязі, мета роботи - досягнута.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Покладіть свої емоції на музику https://wkrolik.com.ua/pokladit-svo%D1%97-emoci%D1%97-na-muziku/ (дата звернення 09.09.2023)

2. Ізометричні перетворення: склад, типи та приклади https://uk.warbletoncouncil.org/transformaciones-isometricas-5104 (дата звернення 12.09.2023)

3. Фадєва К. В. Музичні комп'ютерні технології XX століття. / К. В. Фадєва Монографія. К., 2020, —399 с.

4. Генеративна лінгвістика https://uk.wikipedia.org/wiki/%D0%93%D0%B5%D0%BD%D0%B5%D1%80%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D0%B0_%D0%BB%D1%96%D0%BD%D0%B3%D0%B2%D1%96%D1%81%D1%82%D0%B8%D0%BA%D0%B0 (дата звернення 12.09.2023)

5. Towards melodic extension using genetic algorithms / M. Towsey, A. Brown, S. Wright, S. Diederich. – 2021.

6. Reddin J. Elevated Pitch: Automated Grammatical Evolution of Short Compositions / J. Reddin, J. Mcdermott, M. O'Neill. – 2019.

7. What's a Deep Neural Network? Deep Nets Explained https://www.bmc.com/blogs/deep-neural-network/ (дата звернення 17.09.2023)

8. What is LSTM? Introduction to Long Short-Term Memory https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/ (дата звернення 17.09.2023)

9. A closer look at AI: genetic algorithms https://brighterion.com/artificial-intelligence-101-genetic-algorithms/ (дата звернення 17.09.2023)

10. variational autoencoder (VAE) https://www.techtarget.com/searchenterpriseai/definition/variational-autoencoder-VAE (дата звернення 17.09.2023)

11. Композитор на базі штучного інтелекту. https://www.imena.ua/blog/composer-with-artificial-intelligence/ (дата звернення 17.09.2023)

12. Mubert https://mubert.com/ (дата звернення 22.09.2023)

13. Soundraw https://soundraw.io/ (дата звернення 22.09.2023)

14. AIVA https://www.aiva.ai/ (дата звернення 22.09.2023)

15. Melobytes https://melobytes.com/en/ (дата звернення 22.09.2023)

16. Класифікація нейронних мереж та їх властивості https://dl.nure.ua/pluginfile.php/634/mod_resource/content/2/content/content1.html (дата звернення 27.09.2023)

17. LSTM Recurrent Neural Networks — How to Teach a Network to Remember the Past https://towardsdatascience.com/lstm-recurrent-neural-networks-how-to-teach-a-network-to-remember-the-past-55e54c2ff22e (дата звернення 27.09.2023)

18. Алгоритми конструювання символічної образності пісні https://www.google.com/search?q=%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC+%D0%BA%=&sclient=gws-wiz-serp (дата звернення 05.10.2023)

19. Example of music generation using LSTM networks and Python https://medium.com/@stasinskipawel/example-of-music-generation-using-ltsm-networks-and-python-74cda110a2ad (дата звернення 11.10.2023)

20. Модель рекурентної нейронної мережі для генерації музики / О.С. Комарський, А.Ю. Дорошенко // Проблеми програмування. — 2022. — № 1. — С. 87-93. DOI: https://doi.org/10.15407/pp.2022.01.87

21. Neural Networks and Deep Learning http://neuralnetworksanddeeplearning.com/index.html (дата звернення 11.10.2023)

УДК 004.42

[redacted]

ДОСЛІДЖЕННЯ ПРОЦЕСУ ТРАНСЛЯЦІЇ ВІЗУАЛЬНОГО МИСТЕЦТВА В МУЗИКУ ТА СТВОРЕННЯ КОЛЕКЦІЙ ДЛЯ ЛЮДЕЙ З ВАДАМИ ЗОРУ

У статті досліджується створення музики шляхом автоматизованої генерації звукової композиції за зображенням. Розроблений метод автоматичної генерації звуків за зображенням ґрунтується на спільному використанні нейронних мереж та світломузичної теорії. Результатом роботи стала модель нейронної мережі з двошаровою пакетною LSTM мережею з 512 прихованими нейронами в кожному осередку LSTM.

Ключові слова: *рекурентна нейронна мережа, світломузична теорія, спектрограма, генерація композицій.*

Постановка проблеми

Компанії навчають нейромережі створювати мелодії, пропускаючи через них тисячі пісень у різних жанрах. Штучний інтелект навчають розпізнавати настрій, темп, жанр мелодій, а потім за налаштуваннями створювати нові треки [1].

Найчастіше створення музики за допомогою нейромережі відбувається на основі пісень, які завантажені до її бібліотеки. Деякі нейромережі пишуть ноти, інші створюють композицію з вокалом та купою музичних інструментів. Також існують нейронні мережі, які створюють мелодії за фотографіями. В цьому випадку зображення кодується в музику, а потім реконструюється на основі згенерованого аудіозапису.

Трансляція візуального мистецтва в музику за допомогою моделей машинного навчання може бути використана для створення великих музейних колекцій доступних для людей з вадами зору завдяки переведенню творів мистецтва з недоступної чуттєвої модальності (зір) у доступну (слух).

Аналіз останніх досліджень і публікацій

Першу візуалізацію музики – Atari Video Music – розробив Роберт Браун у 1976 році. Він хотів створити візуалізацію до стереосистеми Hi-Fi [2].

Незважаючи на те, що це було давно, музична візуалізація досі привертає увагу багатьох вчених зі всього світу. З'явились нові технології, такі як генерація музики штучним інтелектом та генерація музичних композицій на основі зображення з використанням алгоритмів глибокого навчання за допомогою зіставлення візуальних, текстових і аудіофункцій.

Дослідження аудіовізуальних моделей показали, що попередні дослідження були зосереджені на покращенні продуктивності моделі за допомогою мультимодальної інформації, а також на покращенні доступності візуальної інформації через аудіоподання.

Мета статті

Мета статті полягає у аналізі методу автоматичної генерації звуків за зображенням, що ґрунтується на спільному використанні нейронних мереж та світломузичної теорії.

Виклад основного матеріалу дослідження

Для зниження ролі користувача-композитора в генерації звуків частина характеристик музичного твору виходить шляхом аналізу кольорової гами зображення. Таким чином, характер отриманої музичної композиції відповідатиме вхідному зображенню. Ця особливість робить можливим застосування цього підходу для створення музейних колекцій доступних для людей з вадами зору.

Ключовими характеристиками музичного твору є його тональність та темп. Саме ці параметри визначаються шляхом аналізу кольорової гами зображення, як показано в табл. 1.

Таблиця 1

Співвідношення кольорових та музичних характеристик [3]

Колірні характеристики	Музичні характеристики
Відтінок (червоний, синій, жовтий…)	Нота (до, до-дієз, ре, ре-дієз, мі, фа, фа-дієз, сіль, сіль-дієз, ля, ля-дієз, сі)
Колірна група (теплий/холодний)	Музичний лад (мажор/мінор)
Яскравість	Октава ноти
Насиченість	Довжина ноти

З табл. 1 робимо висновок, що тональність твору визначається двома кольоровими характеристиками – відтінок та група кольору, а темп – яскравістю та

насиченістю. Алгоритм визначення тональності спирається на аналіз зображення та табл. 1 і складається з чотирьох кроків.

Крок 1. Перетворимо вхідне зображення з простору RGB в HSV. Даний крок дозволяє перетворити зображення до зручнішого вигляду, оскільки HSV-простір вже містить необхідні характеристики: назва кольору (визначається за параметром hue), насиченість (параметр saturation) та яскравість (параметр brightness).

Крок 2. Аналізуючи загалом зображення, визначаємо переважний колір.

Крок 3. Визначаємо назву та групу кольорів переважного кольору.

Крок 4. Згідно з табл. 1 та схемою Ньютона визначаємо тональність твору (ноту та музичний лад).

Для визначення темпу твору необхідно отримати яскравість і насиченість (за параметрами saturation і brightness) переважного кольору і розрахувати темп, згідно з даними параметрами (рис. 1).

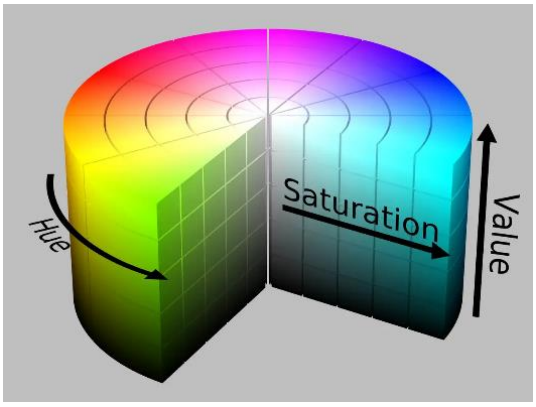


Рис. 1. Циліндр HSV

Результатом цих кроків є графічна анотація перетворення графічного зображення в музикальний ряд з використанням всіх кольорових характеристик, яка передається на вхід нейронної мережі.

Нейронна мережа з математичної точки зору поводиться як звичайна функція, хоч і дуже складно влаштована. Вона має заздалегідь позначену кількість аргументів і позначений формат, в якому вона видає відповідь.

В роботі було використано LSTM мережу, тому що вона має пам'ять про стан осередку і можуть переносити інформацію про більш довгострокові структури в музиці порівняно з рекурентною нейронною мережею (RNN) та мережею з архітектурою керованого нейрона (GRU), що дозволяло нам передбачати довші послідовності до 1 хвилини, які очікувано будуть звучати узгоджено.

Для реалізації програми автоматизованої генерації музичних композицій за зображенням є сенс використовувати рекурентні нейронні (RNN) мережі з довготривалою пам'яттю (LSTM) [4], які намагаються вирішити проблему звичайних RNN мереж – втрату інформації з часом, використовуючи фільтри та явно задану клітину пам'яті. Метою цих фільтрів є захист інформації. Вхідний фільтр визначає, скільки інформації з попереднього шару зберігатиметься в нейроні. Вихідний фільтр визначає, скільки інформації отримають такі шари [5].

Основна ідея вирішення задачі генерації музики за зображенням представляє собою складання спектрограм за вхідним рядком зображення і конвертації її в аудіокліп.

Аудіо спектрограма – це візуальний спосіб представлення частотного змісту звукового кліпу [6]. Вісь X є час, а вісь Y є частота (рис. 2). Колір кожного пікселя визначає амплітуду звуку залежно від частоти та часу.

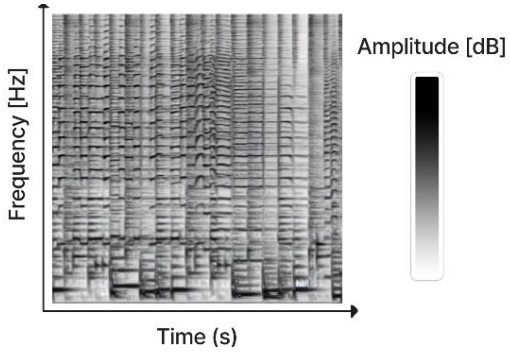


Рис. 2. Спектрограма звукового кліпу

Спектрограма може бути отримана зі звуку з використанням перетворення Фур'є (STFT), яке апроксимує звук як комбінацію синусоїдальних хвиль різної амплітуди та фази (рис. 3).

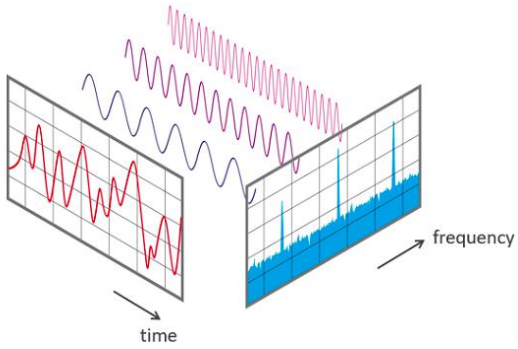


Рис. 3. Спектрограма як комбінація синусоїдальних хвиль різної амплітуди та фази [7]

У процесі дослідження методів синтезу звуків було розглянуто та проаналізовано найбільш популярні методи: адитивний синтез, FM-синтез, фазова модуляція, семплінг, таблично-хвильовий синтез, лінійно-арифметичний синтез, субтрактивний синтез та векторний синтез [8].

FM-синтез добре застосовний для синтезу звуку ударних інструментів, синтез інших музичних інструментів звучить занадто штучно. Головний недолік FM-синтезу – нездатність за його допомогою повноцінно імітувати акустичні інструменти.

Фазова модуляція дає досить гарний звук, але дуже обмежена за своїми можливостями, тому рідко використовується на практиці.

Семплінг застосовується в більшості сучасних синтезаторів, тому що дає найбільш реалістичний звук та досить простий у реалізації [9].

Таблично-хвильовий синтез та лінійно-арифметичний синтез схожі на метод семплінгу, але вони складні в реалізації, тому на практиці перевага віддається семплінгу як найпростішому методу.

Субтрактивний синтез зазвичай використовується спільно з адитивним, має хорошу якість синтезу звуків, проте складний у реалізації.

Векторний синтез використовується для отримання більш багатих і складних тембрів, однак у рамках розгляду системи це не суттєво.

Тому для реалізації системи було обрано саме метод семплінгу. Цей метод дає найбільш реалістичне звучання інструментів, що є важливою її характеристикою.

Для реалізації запропонованих алгоритмів генерації звуків за кольоровою гамою зображення було розроблено базову модель нейронної мережі з двошаровою пакетною LSTM мережею з 512 прихованими одиницями в кожному осередку LSTM [10]. Ми використовували шар впровадження, щоб перетворити кожен токен (усереднене значення кольору) у вектор (рис. 4).

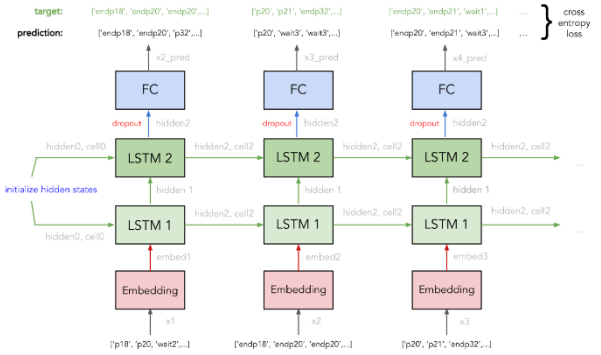


Рис. 4. Архітектура LSTM мережі

Програму генерації музичних композицій з використанням нейронних мереж було навчено на 29 композиціях сучасної музики.

Гіперпараметри що були налаштовані у LSTM мережі, такі як наприклад визначення довжини послідовності для вибірки, не повинні бути занадто короткими бо їх довжини буде не достатньо для того, щоб створити музичний ланцюжок, який звучить узгоджено, але й занадто довга послідовність займуть надто багато часу для навчання мережі без вивчення додаткової інформації.

Після навчання було складено набір із десяти тестових зображень, що мають різний тип (абстрактні зображення, пейзажі, міста та люди), за якими були отримані та збережені вихідні музичні композиції. Всі музичні композиції були оцінені за такими критеріями:

– відповідність характеру зображення (за п’ятибальною шкалою);
– реалістичність звучання інструменту (фортепіано чи гітара);
– мелодійність композиції;
– якість гармонії (акомпанементу);
– приємність мелодії для сприйняття;
– цілісність композиції;
– реалістичність/штучність композиції.

За результатами оцінки по всім тестовим зображенням було розраховано середні значення, які представлені у табл. 2.

Таблиця 2

Оцінка композицій за критеріями

Критерій	Середнє значення для всіх тестів
Відповідність характеру зображення	4.9
Реалістичність звучання інструмента	3.9
Мелодійність композиції	4.4
Якість гармонії	4.9
Приємність для сприйняття	4.6
Цілісність композиції	4.5
Реалістичність композиції	4.3

Ми використовували метрику BLEU, яка оцінює кількість n-грам (повтори з вихідних даних), що використовуються в згенерованих послідовностях для порівняння нашої базової моделі LSTM із деякими реальними композиціями. Результати - усереднені по тестовим зображенням послідовностей - 0,25 для LSTM та 0,14 для реальних даних. Це означає, що наші згенеровані зразки хороші за нашими налаштованими метриками. Вища оцінка, яку композиції LSTM дають порівняно з реальними композиціями, підкреслює наявність переоснащення де згенерована музика містить суттєві повтори з вихідних даних.

Загалом можна зробити висновок, що композиція, згенерована за абстрактним зображенням, приємніша на слух, ніж генерація за пейзажами. У цілому загальне враження від

згенерованих композицій позитивне. Серед мінусів слід звернути увагу на однотипність гармонії, іноді рваність та недостатню реалістичність твору.

Висновки

У результаті виконання цього дослідження було запропоновано комбінований підхід до генерації звукових послідовностей. Він використовує рекурентну нейронну мережу для генерації музичного матеріалу та кольорову музичну теорію для визначення параметрів композиції із зображення. У процесі вибору нейронної мережі для генерації музичних композицій було виявлено, що для реалізації програми автоматизованої генерації музичних композицій необхідно використовувати саме рекурентні нейронні мережі з довготривалою пам'яттю – RNN LSTM.

Результати роботи можуть бути використані для створення великих музейних колекцій доступних для людей з вадами зору завдяки переведенню творів мистецтва з недоступної чуттєвої модальності (зір) у доступну (слух).

Література

1. Як штучний інтелект створює музику і змінює креативну індустрію [Електрон. ресурс] / Depositphotos : сайт. – США, 2009–2023. – Оновлюється постійно. – Режим доступу: https://blog.depositphotos.com/ua/yak-shtuchnyj-intelekt-stvoryuye-muzyku.html, вільний (дата звернення: 05.10.2023).

2. GANSynth: Adversarial Neural Audio Synthesis / J. Engel, K. K. Agrawal, S. Chen, I. Gulrajani, C. Donahue, A. Roberts // Proceedings of the 7th International Conference on Learning Representations (ICLR), New Orleans, LA (USA), May 6–9, 2019 yr. – 17 p. – DOI: 10.48550/arXiv.1902.08710.

3. Caivano J. L. Color and Sound: Physical and Psychophysical Relations / J. L. Caivano // Color Research and Application. – 1994. – Vol. 19 (2). – P. 126–133. – DOI: 10.1111/j.1520-6378.1994.tb00072.x.

4. Комарський О. С. Модель рекурентної нейронної мережі для генерації музики / О. С. Комарський, А. Ю. Дорошенко // Проблеми програмування. – 2022. – № 1. – С. 87–93. – DOI: 10.15407/pp.2022.01.87.

5. A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck. A hierarchical latent vector model for learning long-term structure in music. arXiv:1803.05428v5 [cs.LG] 11 Nov 2019, 2019. https://arxiv.org/pdf/1803.05428.pdf

6. Яровий М. В. Частотний аналіз в задачах розпізнавання звуку з використанням нейроних мереж / М. В. Яровий, О. С. Назаров // Сучасні аспекти та перспективи розвитку науки : матеріали І міжнар. студ. наук. конф., Кропивницький, Україна, 16 квітня 2021 р. – Кропивницький : Молодіжна наукова ліга, 2021. – Т. 2. – С. 48–50. – Режим доступу: https://ojs.ukrlogos.in.ua/index.php/liga/issue/view/16.04.2021/502, вільний (дата звернення: 05.10.2023).

7. Бондаренко А. І. Виявлення і аналіз акустичних подій в електронній музиці (на прикладі "Мотус" А. Загайкевич) / А. І. Бондаренко // Питання культурології. – 2015. – Вип. 31. – С. 22–28. – Режим доступу: http://nbuv.gov.ua/UJRN/Pkl_2015_31_5, вільний (дата звернення: 05.10.2023).

8. Кущ Є. В. Про деякі аспекти функціонування електромузичного інструментарію у музичній культурі другої половини XX століття / Є. В. Кущ // Наукові записки Тернопільського національного педагогічного університету імені Володимира Гнатюка. Серія: Мистецтвознавство. – Тернопіль: Вид-во ТНПУ ім. В. Гнатюка, 2013. – № 1. – С. 17–23. – Режим доступу: http://dspace.tnpu.edu.ua/bitstream/123456789/3824/1/KUSHCH.pdf, вільний (дата звернення: 05.10.2023).

9. How to Sample Music: Step-by-Step Music Sampling Guide – Оновлюється постійно. – Режим доступу: https://masterclass.com/articles/how-to-sample-musicml, вільний (дата звернення: 05.10.2023).

10. Alekseev, P.P. and Kviatkovskaia, I.Iu., 2021. Primenenie neironnykh setei dlia raspoznavaniia printcipialnykh uslovno-graficheskikh elektricheskikh oboznachenii [The use of neural networks for the recognition of fundamental conventional graphic electrical symbols]. Vestnik Astrakhanskogo gosudarstvennogo tekhnicheskogo universiteta. Seriia: Upravlenie, vychislitelnaia tekhnika i informatika, [online] 2, pp.47-56. – Режим доступу: http://www.mathnet.ru/links/57f0ef0c3d69c2f4645481ceae10f0f4/vagtu669.pdf