

Санкт-Петербургский государственный университет

Умути Сайжикэцзян

Выпускная квалификационная работа

Нейросетевые методы генерации музыкальных дорожек

Уровень образования: магистратура

Направление *02.04.02 «Фундаментальная информатика и информационные технологии»*

Основная образовательная программа *ВМ.5503.2021 «Технологии баз данных»*

Научный руководитель:

Доцент кафедры технологии программирования, к. т. н. И. С. Блеканов

Рецензент:

Руководитель проектного офиса Департамент цифровой трансформации Skillbox
Я. С. Кононов

Санкт-Петербург
2023

Saint Petersburg State University

Wumuti Sairikejiang

Master's Thesis

Neural network methods for generating music tracks

Education level: master

Speciality *02.04.02 «Fundamental Informatics and Information Technology»*

Programme *BM.5503.2021 «Database Technologies»*

Scientific supervisor:
C.Sc., docent I. S. Blekanov

Reviewer:
Head of Project Office Digital Transformation Department Skillbox Y. S. Kononov

Saint Petersburg
2023

Оглавление

Введение	4
1. Обзор предметной области	8
1.1. Метод представления и кодирования музыки	8
1.1.1. Метод представления	8
1.1.2. Метод кодирования	10
1.2. Статус исследования генерации музыки	11
1.3. Задача генерации музыки	14
1.3.1. Генерация мелодий	14
1.3.2. Гармоническая аранжировка	15
1.3.3. Перенос стиля	16
1.4. Классическая модель	17
1.4.1. Модель LSTM	17
1.4.2. Модель Transformer	19
1.4.3. Модель GAN	21
2. Классическая и усовершенствование модель	22
2.1. Классические модели	23
2.1.1. Генерация музыки на LSTM и Transformer	23
2.1.2. Генерация музыки на MuseGAN	27
2.2. Классическая модель и ее усовершенствование	29
2.2.1. Усовершенствованные модели LSTM	29
2.2.2. Усовершенствованные модели GAN	32
2.3. Субъективная оценка результатов	37
3. Заключение	42
Список литературы	44

Введение

Актуальность

С развитием компьютерных технологий и распространением электронных устройств музыка все чаще хранится в цифровом виде на компьютерах, а хранение, обработка, сочинение и производство музыки все больше зависят от компьютеров. Эта тенденция определила развитие музыкальных технологий и музыкального интеллекта. Если музыка - это искусство и культурная форма, используемая для выражения мыслей и чувств людей и социальных реалий, то генеративный искусственный интеллект - одна из самых революционных технологий нашего времени, меняющая способ нашего взаимодействия с машинами. Генерация музыки изучает идею о том, чтобы позволить компьютерам автоматически создавать музыку с помощью алгоритмов, что обеспечивает музыкальной индустрии более разнообразный спектр музыкальных произведений для удовлетворения потребностей различных аудиторий.

В последние годы глубокие нейронные сети [4] добились значительного прогресса и достигли замечательных результатов в генерации изображений и видео, а также текста [37]. Аналогичные попытки постепенно предпринимаются для применения их в задачах генерации музыки. С развитием методов глубокого обучения [5] применение методов глубокого обучения [25] для автоматического генерирования музыки стало актуальным направлением исследований. Методы глубокого обучения позволяют избежать большого количества ручных музыкальных характеристик, которые необходимо маркировать в традиционных алгоритмах машинного обучения, и стали основным подходом к автоматическому генерированию музыки, обусловленным текущими доступными музыкальными данными и хорошими возможностями моделирования. В настоящее время генерация музыки охватывает различные области, такие как генерация полифонической музыки [19], генерация аккомпанемента [34], преобразование стиля [22] и восстановление музыки [27].

Создание музыкального произведения - непростая задача, это не то,

что может сделать один человек простым движением руки, скорее, композитору требуется много усилий и времени, чтобы тщательно разработать и создать музыкальное произведение с помощью человеческого труда.

С развитием технологий искусственного интеллекта генерация музыки стала одним из основных направлений исследований в области искусственного интеллекта. Использование нейронных сетей для генерации музыки позволяет автоматически генерировать музыку, снижая порог для сочинения треков без необходимости для создателя обладать большим количеством специальных знаний в области теории музыки; все, что требуется, - это четкое представление о том, чего он хочет, и компьютер может сам создать соответствующий трек. Это не только помогает музыкантам в их творческой работе, давая вдохновение и создавая уникальные музыкальные произведения, но и обеспечивает музыкальную индустрию более разнообразным ассортиментом музыкальных произведений, отвечающих потребностям различных аудиторий. В то же время технология генерации музыки может доставить любителям удовольствие от написания и сочинения песен. Генерация фортепианной музыки является одним из наиболее важных применений технологии генерации музыки и имеет широкое исследовательское и практическое значение.

Цель работы

Целью работы является исследование методов генерации фортепианной музыки с использованием нейронных сетей, таких как LSTM, Transformer и GAN, обучение нескольких моделей генерации фортепианной музыки и оценка их производительности. Кроме того, в данной работе оценивается субъективное качество сгенерированных музыкальных клипов с помощью анкеты для выбора лучшей модели генерации фортепианной музыки, а также разрабатывается и внедряется приложение для автоматической генерации музыки на основе этой модели.

Задачи работы

Для достижения цели необходимо:

1. Сбор и подготовка данных, используемых для проведения экспериментов, и их предварительная обработка.
2. Обучение моделей генерированию одноподорожечных и многодорожечных музыкальных клипов для фортепиано и выбор оптимального клипа для каждой модели.
3. Разработка анкеты субъективной оценки для распространения среди целевой аудитории и анализ полученных результатов.
4. Разработать программу автоматической генерации музыки LSTM-композитор на основе лучшей модели генерации фортепианной музыки, описав структуру и функциональность программной системы.

Практическая значимость работы

В работе исследуются методы генерации фортепианной музыки с использованием нейронных сетей, таких как LSTM, Transformer и GAN, для обучения пяти моделей генерации фортепианной музыки, а также приводится открытый исходный код этих пяти моделей. Кроме того, в данной работе оценивается субъективное качество сгенерированных музыкальных фрагментов с помощью анкеты, приводится образец анкеты, выбирается лучшая модель генерации фортепианной музыки, разрабатывается и внедряется приложение для автоматической генерации музыки на основе этой модели для генерации фортепианных музыкальных произведений.

Выводы, сделанные в данной работе, имеют большое практическое значение. Во-первых, технология генерирования фортепианной музыки может обеспечить больше творческого вдохновения для авторов фортепианной музыки и больше учебных ресурсов и возможностей для практики для учащихся музыкальных школ, помогая им лучше овладеть

знаниями и навыками теории музыки. Во-вторых, фортепианная музыка, как элегантный вид искусства, обладает успокаивающим эффектом и может принести людям внутренний покой и расслабление. Генерирование большего количества произведений классической музыки с помощью технологии генерации музыки может не только предоставить больше возможностей для любителей музыки оценить и научиться, но и позволить большему количеству людей найти эмоциональную поддержку и духовное утешение в прекрасном мире классической музыки. Кроме того, созданные фортепианные пьесы и отрывки могут быть использованы в области музыкального образования, музыкальной терапии и музыкальной композиции, расширяя спектр музыкальных приложений. Наконец, развитие технологии генерации музыки также принесло новые бизнес-модели и возможности в музыкальную индустрию, способствуя дигитализации и интеллектуальному развитию музыкальной индустрии. Таким образом, результаты исследования, проведенного в данной работе, имеют большое практическое значение для развития музыкальной индустрии и продвижения классической музыки.

1. Обзор предметной области

1.1. Метод представления и кодирования музыки

1.1.1. Метод представления

Музыка как художественный язык не может быть непосредственно понята компьютером, поэтому символическую музыку необходимо характеризовать, чтобы облегчить сетевым архитектурам глубокого обучения извлечение и изучение особенностей данных. Символическая музыка в основном характеризуется событиями MIDI, фортепианными записями, текстом и другими методами представления. В зависимости от архитектуры модели и стратегии ее генерации, выбор различных методов характеристики оказывает важное влияние на производительность модели и качество генерируемой музыки.

MIDI-Like

MIDI использует события Note on и Note off для представления начала и конца ноты, соответственно, а высота тона ноты представлена целым числом от 0 до 127, в дополнение к каждому событию ноты вложена структура данных, содержащая значение Delta-time, которое определяет временную информацию относительно времени.¹ Пример представления MIDI показан на рисунке 1 где первая линия представляет середину C шестнадцатой ноты, сыгранной на первом канале со скоростью 90.

MIDI-Like представление в настоящее время наиболее широко используется в задачах генерации символьной музыки на основе глубокого обучения, но оно только записывает процесс воспроизведения музыки и не обеспечивает визуального представления содержания.

Piano roll

Piano roll – это двухмерная таблица с непрерывными временными шагами на горизонтальной оси и высотой тона на вертикальной оси.² Пример представления Piano roll показан на рисунке 2, который обеспечивает более интуитивное представление высоты тона и длительности нот, но не других музыкальных свойств, таких как интенсивность и

1,	96,	Note_on,	0,	60,	90
1,	288,	Note_off,	0,	80,	0
1,	192,	Note_on,	0,	60,	90
1,	96,	Note_off,	0,	60,	0
1,	384,	Note_on,	0,	80,	90
2,	96,	Note_off,	0,	80,	0
2,	288,	Note_on,	0,	84,	90
2,	96,	Note_off,	0,	84,	0

Рис. 1: MIDI

тембр.

Текстовые представления

Основными текстовыми представлениями являются представление ABC и представление MusicXML. Представление ABC является наиболее широко используемым, оно было впервые применено для традиционной и народной музыки на таких языках, как английский и ирландский, а сейчас используется в широком спектре музыкальных жанров.

ABC-представление описывает основную информацию о названии, темпе, высоте тона и высоте штриха музыкального произведения. Пример представления ABC показан на рисунке.

MusicXML4 является стандартным документом в области нотной музыки и музыкальной электронизации. В 2004 году компания Recordare выпустила MusicXML версии 1.0, в которой для описания нотной музыки используется формат документа XML. В настоящее время версия 3.1 доступна для всех партитур классической музыки, начиная с 18 века, и может представлять такую информацию, как: полифония, несколько инструментов, высота тона, восходящая и нисходящая, пентатоническая позиция, интенсивность и т.д. По сравнению с MIDI-Like, представление MusicXML позволяет более эффективно записывать логику исполнения и более наглядно представлять партитуру. Однако, в силу своей совместимости, представление партитур является более сложным

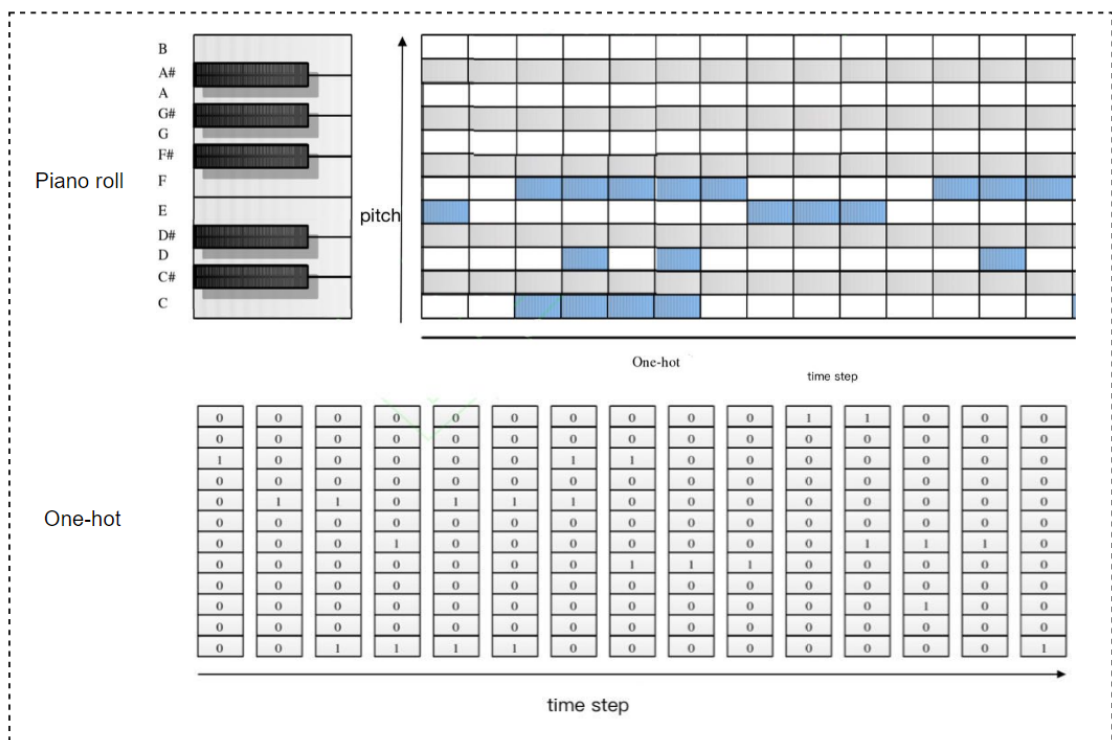


Рис. 2: Представление Piano roll и кодирование One-hot.

и может значительно увеличить внутреннюю работу сети, влияя на эффективность модели.

1.1.2. Метод кодирования

Для того чтобы лучше извлечь особенности музыки, за выбором соответствующего представления следует выбор подходящего метода кодирования музыки. Кодирование музыки заключается в отображении представления (состоящего из набора переменных, таких как высота тона или динамика) на набор входов (также называемых входными узлами или входными переменными) архитектуры нейронной сети [3, 23]. Существует два основных подхода: Value Encoding [2] и One Hot Encoding, при котором значение кодируется непосредственно как скаляр через непрерывные, дискретные или булевы переменные. One Hot Encoding кодирует дискретную или категориальную переменную как категориальную переменную посредством вектора, длина которого равна числу всех возможных элементов. Пример Solo Hot Encoding, соответствующего представлению Piano roll, показан на рис 3.

X: 24
 T: Clouds Thicken
 C: Paul Rosen
 S: Copyright 2005, Paul Rosen
 M: 6/8
 L: 1/8
 Q: 3/8=116
 R: Creepy Jig
 K: Em
 l:"Em"EEEE2G|"C7"_B2AG2F|"Em"EEEE2G|"C7"_B2A"B7"=B3|"Em"EEE
 E2G|"C7"_B2AG2F|"Em"GFE"D(Bm7)"F2D|1"Em"E3-E3:l2"Em"E3-
 E2B|:"Em"e2egfel"G"g2ab3|"Em"gfe g2el"D"fedB2A|"Em"e2egfel"G"g2ab3|"Em

Рис. 3: ABC-представление.

1.2. Статус исследования генерации музыки

Использование компьютеров для сочинения музыки имеет продолжительную историю. В 1956 году Пинкертон и другие ученые [28] впервые попытались генерировать музыкальные мелодии с помощью компьютера. В том же году в Университете штата Иллинойс в Урбане-Шампейне была сгенерирована первая музыкальная композиция, сочиненная компьютером, "Иллиак-сюита"(The Illiac Suite), заложившая основу компьютерной композиции. С тех пор было проведено множество исследований в области алгоритмической композиции.

Традиционно алгоритмическая композиция делится на три основные категории: первая категория - генерация музыки на основе правил [30], где используются определенные правила или грамматики для создания музыки. Исследователям необходимо создавать соответствующие правила для различных типов музыки, и этот процесс композиции обычно требует множество ручных настроек и вмешательств, что приводит к относительной неслушабельности, поэтому генерация требует большого количества ручных корректировок на этапе постобработки. Второй тип генерации музыки основан на вероятностных моделях, включая модели Маркова [32] и скрытые модели Маркова [33]. Однако такие модели, как правило, не обладают памятью, что может привести к отсутствию целостной структуры в сгенерированной музыке, а также к повторению фрагментов из обучающего датасета [24].

Последняя категория использует глубокие нейронные сети для изу-

чения характеристики музыки и генерации музыкальный фрагментов на основе этих изученных характеристик.

С 1980-х годов до начала 21 века исследователи начали использовать нейронные сети для композиции. Например, Тодд и др [35]. впервые использовали рекуррентные нейронные сети [9] для генерации мелодии, предсказывая высоту тона и длительность нот. Однако ранние RNN-модели столкнулись с проблемами исчезновения и взрыва градиентов при обработке длинных последовательностей, что затрудняло создание музыки с долгосрочной структурой. С непрерывным развитием и оптимизацией методов глубокого обучения многие исследователи пытались моделировать музыку с помощью глубоких нейронных сетей, в следствие чего генерация музыки на основе глубокого обучения постепенно становится оптимальным подходом для современных исследований.

Сочинение музыки – это сложный процесс, который опирается на большое количество композиционных правил и техник. Из-за сложности временной структуры музыкальных последовательностей и их высокохудожественного характера, генерация музыки на основе глубокого обучения также сталкивается с особыми вызовами, и исследователи обычно применяют специальные подходы и методы в зависимости от поставленных задач.

Цель - использовать алгоритмические методы, а также музыковедение, чтобы помочь компьютерам изучать музыкальные особенности и генерировать музыку, необходимую человеку. Основные шаги показаны на рисунке 4. Во-первых, конкретный набор данных символической музыки характеризуется как компьютерный язык; во-вторых, особенности данных абстрагируются; и, наконец, выбирается подходящая архитектура модели и обучается генерировать символическую музыку в виде музыкальной партитуры.

В современных исследованиях генерация музыкальных дорожек включает четыре основные категории, безусловная генерация, генерация продолжения, генерация восстановления, генерация атрибутов и контроля, последние три из которых могут быть классифицированы как условная генерация в противоположность безусловной генерации. Без-

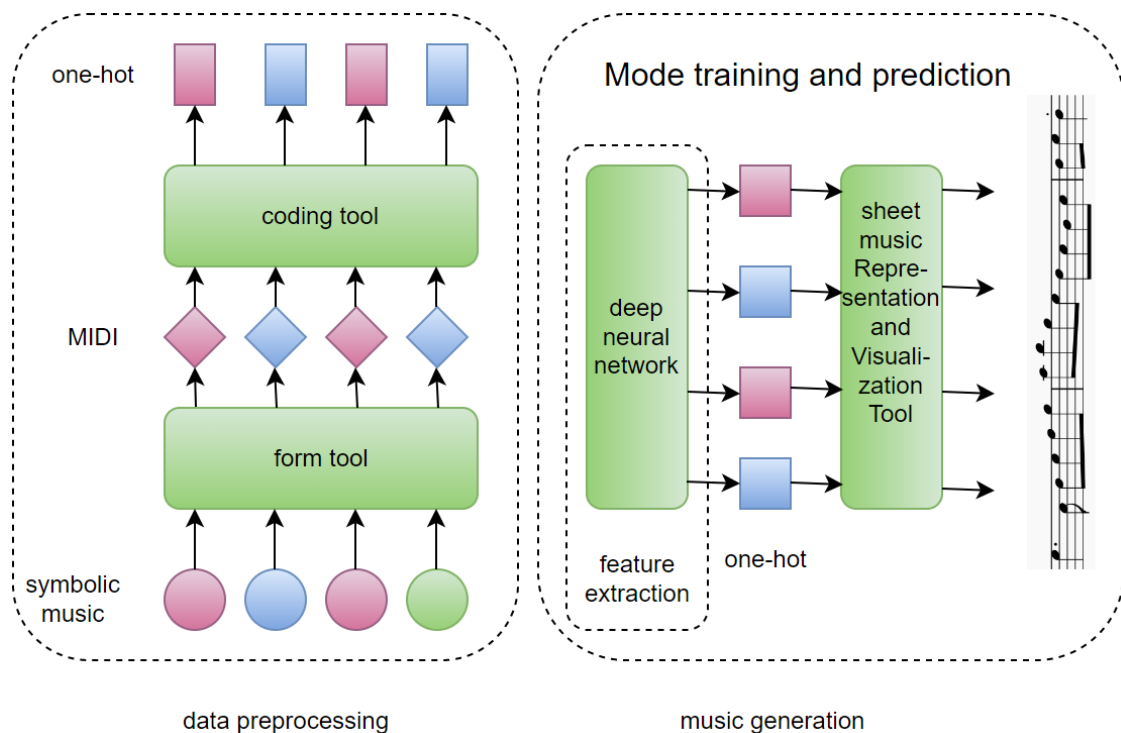


Рис. 4: Основные этапы генерации музыки.

условная генерация – это генерация новой музыки без вмешательства человека исходя из изученных музыкальных характеристик в наборе данных; Продолжение генерации – это точная настройка модели на конкретном музыкальном материале для продолжения незаконченного музыкального произведения; генерация восстановления предполагает заполнение недостающих нот, тактов, дорожек и так далее в музыкальном материале; атрибутно-управляемая генерация – это генерация музыкальных фрагментов определенного стиля, ритма и с учетом других атрибутов, кроме того может быть управляема пользователем, что удовлетворяет взаимодействию между пользователем и генеративной системой и имеет большую практическую ценность в реальных сценариях.

С художественной точки зрения, сгенерированная музыкальная композиция должна обладать такими атрибутами, как характерная тема, гармоничная мелодия, последовательная структура, соответствующая гармоническому сопровождению и стабильный ритм. Более того, эти атрибуты взаимосвязаны, например, тема музыки влияет на мелодическое направление и гармоническую прогрессию; мелодическое и гармония музыки влияют на структурную организацию музыки; гармониче-

ская прогрессия имеет развитие или завершение определенной структуры. Поэтому генерация музыки на основе глубокого всестороннего исследования с упором на вышеперечисленные аспекты.

1.3. Задача генерации музыки

Исследуемые задачи по генерации музыки на основе глубокого обучения включают три основные категории: генерация мелодии, подбор гармоник и перенос стилей.

1.3.1. Генерация мелодий

Музыкальное творчество обычно начинается с четкой темы, которая затем выражается через мелодию. Генерация мелодии – важная задача в компьютерной композиции, учитывающая такие атрибуты, как согласованность музыкальной структуры, длину последовательности и другие.

Строго говоря, каждое музыкальное произведение имеет свою уникальную структуру, и после анализа и классификации большого количества музыкальных произведений, основными типами мелодической структуры являются одночастная, одночастная двухчастная, составная трехчастная и сонатная формы. Содержание музыки определяет ее структуру, а структура влияет на выражение темы, поэтому генерация музыкальных мелодий со структурой находится в центре внимания исследователей в области генерации музыки на основе глубокого обучения, а также является основной задачей в текущей работе.

В настоящее время исследование музыкальной структуры в области генерации символьной музыки на основе глубокого обучения сосредоточено на двух основных аспектах: изучение алгоритмов моделирования и более эффективное моделирование структуры. В контексте исследования моделей и алгоритмов, в 2015 году Латтнер и др [29]. предложили использовать комбинацию ограниченной машины Больцмана со сверточными слоями в качестве генеративной модели в сочетании с алгоритмом оптимизации градиентного спуска для контроля процесса гене-

рации, используя при этом структуру в качестве ограничения и передавая ее вновь сгенерированной музыке, создавая музыку с более полной структурой. Что касается структурного моделирования, то в 2018 году Чен и др [8]. получили дальнейшие улучшения в плане композиционной структуры генерируемой музыки за счет явного кодирования структуры.

В ранних работах исследователи в основном использовали статистические методы, основанные на данных, такие как модели N-грамм [21] и модели Маркова. Позднее Гоэль и др [13]. использовали рекуррентные нейронные сети (RNN) для генерации мелодии. Однако эти методы не учитывали иерархическую структуру и семантические отношения в музыке, что затрудняло генерирование приятных слуху и разнообразных мелодических последовательностей мелодий. С появлением модели долгой краткосрочной памяти [15] (LSTM) качество генерации мелодии в целом значительно улучшилось. В 2016 году Бретан и др. использовали семантические единицы и сети LSTM для захвата иерархической структуры музыки. Хотя был достигнут определенный прогресс в решении проблемы временной структурированности, генерируемые мелодические последовательности были относительно короткими. Для исправления этих недостатков ученые пробовали построить новые модели генерации музыки для исследований, и репрезентативные модели генерации в этот период были в основном основаны на нейронных [1, 12, 18], таких как вариативные автоэнкодеры и генеративные адверсарные сети в качестве архитектур. Среди них модели на основе трансформеров совершили прорыв с точки зрения длины последовательности и структурной согласованности генерируемой музыки. В настоящее время для получения лучших результатов часто используются комбинации этих моделей, такие как MusicVAE [14], Transformer VAE [36], PianoTree [26] и др.

1.3.2. Гармоническая аранжировка

Гармония является существенным элементом музыки, представляя собой связь между мелодией и аккордами или между аккордами, и име-

ет две основные функции: построение структуры и выражение цвета. Основная задача гармонической аранжировки - генерировать аккордовые соединения различных цветов в соответствии с мелодией. В работах по генерации музыки на основе глубокого обучения генерация аккордов была относительно мало изучена по сравнению с генерацией мелодий, а модели в основном изучались на основе сетей RNN и их разновидностей.

В 2016 году Эк и др [7]. впервые сгенерировали последовательности аккордов с помощью LSTM-сетей, и аналогично Лим и др [20]. В 2019 году Ян и др [6]. улучшили эту модель, предложив модель CLSTMS для генерации аккордов, соответствующих мелодии, которая состоит из двух LSTM, одна из которых изучает связь между мелодией и аккордом, а другая передает правила преобразования аккордов в модель. В 2021 году Чен и др [38]. провели аналогичную работу, предложив модель GCA, основанную на структуре двух LSTM. Первая используется для генерации мелодий, а вторая объединяет сгенерированные мелодии для генерации аккордов. Основываясь на этих исследованиях, Ву и др [11]. предложили модель Auto Harmonizer для согласования мелодии с соответствующей гармонией. Модель состоит из двух модулей: гармонического ритма, который предоставляет крупнозернистую информацию о начале аккорда, и высоты аккорда, которая определяется на основе заданной мелодии и сгенерированного гармонического ритма.

1.3.3. Перенос стиля

Стиль музыки определяется множеством элементов, включая ритм, ключ, мелодию и гармонию музыки. Композитор может расположить эти элементы определенным образом, чтобы получить целевой музыкальный стиль. Однако для компьютера получение конкретного музыкального стиля требует извлечения стилистических особенностей из соответствующего набора данных [10]. В 2018 году Бруннер и др. предложили модель MIDI-VAE - первое применение миграции стилей для генерации символьной музыки. Используя высоту тона, ритмическую

динамику и инструментальные комбинации в качестве стилистических особенностей, MIDI-VAE позволяет модели более эффективно изучать эти особенности и генерировать музыкальные композиции в заданном стиле.

Кроме того, цель генерирования музыки, специфичной для конкретного жанра, может быть достигнута с помощью методов трансферного обучения, когда модель сначала предварительно обучается на наборе данных [17], специфичном для конкретного жанра, а затем дорабатывается на целевом наборе данных.

1.4. Классическая модель

В задаче создания символической музыки текущие основные генеративные модели в основном используют LSTM, Transformer и GAN в качестве базовой архитектуры.

1.4.1. Модель LSTM

LSTM – это усовершенствованная сетевая архитектура на основе RNN, которая ранее использовалась для задач с временными рядами и широко применялась в обработке естественного языка, а также в задачах генерации музыки [31]. Однако RNN не способны эффективно запоминать информацию в последовательностях, находящихся на большом расстоянии друг от друга, с одной стороны, и не способны избирательно запоминать информацию в длинных последовательностях, с другой стороны, что приводит к тому, что музыкальные последовательности менее связны и не имеют более однородной структуры.

LSTM устраняет вышеупомянутые недостатки традиционных RNN путем добавления ячеек памяти и механизмов стробирования. LSTM совершенствуется путем добавления ячеек памяти и механизмов стробирования, чтобы уменьшить нагрузку на память и тем самым улучшить общую производительность. Базовая архитектура LSTM показана на рисунке ???. Включены ячейки LSTM в моменты времени $t-1$, t и $t+1$. где s_t представляет выход состояния ячейки памяти в момент време-

ни t , h_t представляет выход скрытого состояния в момент времени t , x_t представляет входной вектор в момент времени t . Механизм управления состоит из трех ворот: ворота забывания, ворота обновления и ворота выхода. Затвор забывания используется для принятия решения об отбрасывании информации из текущего состояния ячейки; затвор обновления обновляет состояние ячейки и состояние ячейки входной памяти на основе входного вектора и входного скрытого состояния в текущий момент времени; а затвор выхода используется для определения выходного скрытого состояния текущей ячейки LSTM.

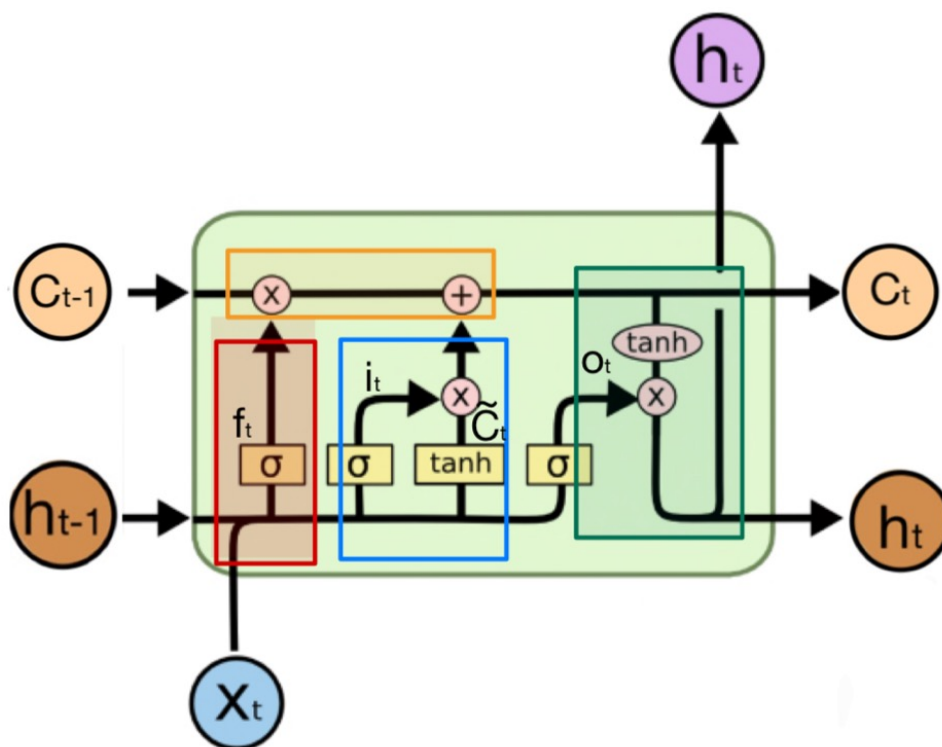


Рис. 5: Архитектурная схема LSTM.

Формула расчета показана ниже:

forget gate:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

output gate:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(C_t)$$

input gate:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$
$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_f)$$

cell state:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

1.4.2. Модель Transformer

Transformer использует базовую архитектуру кодера-декодера, где компоненты кодера и декодера укладываются в стек с одинаковым количеством кодиров и декодеров соответственно. Базовая архитектура Transformer показана на рисунке 6. Блок кодера состоит из двух компонентов: многоголового механизма внимания и нейронной сети с прямой передачей. Подобно блоку кодера, блок декодера также содержит многоголовый механизм внимания и нейронную сеть с прямой передачей, но с дополнительным слоем многоголового механизма внимания по сравнению с кодером, чтобы сосредоточиться на соответствующих частях входной последовательности. В то же время к блокам кодировщика и декодировщика добавляются остаточные связи, обозначенные на рисунке символом Add, чтобы избежать проблем деградации во время обучения глубокой нейронной сети. Затем следует процесс нормализации слоев для ускорения сходимости, обозначенный на рисунке как Norm.

Механизм самовнимания является основным механизмом transformer, цель которого - отфильтровать более эффективную информацию. Улавливая корреляции между векторами и вычисляя их веса внимания в каждой позиции в процессе кодирования, получается неявное векторное представление всей последовательности в виде суммы весов. Процесс вычисления можно описать как отображение вектора запроса и серии пар векторов ключ-значение на выход. Выходной вектор представляет собой сумму весов, примененных к Value на основе весов, вычисленных по Query и Key. Формула расчета показана ниже. где Q, K, V представляют собой произведение указанных векторов и соответствующих им

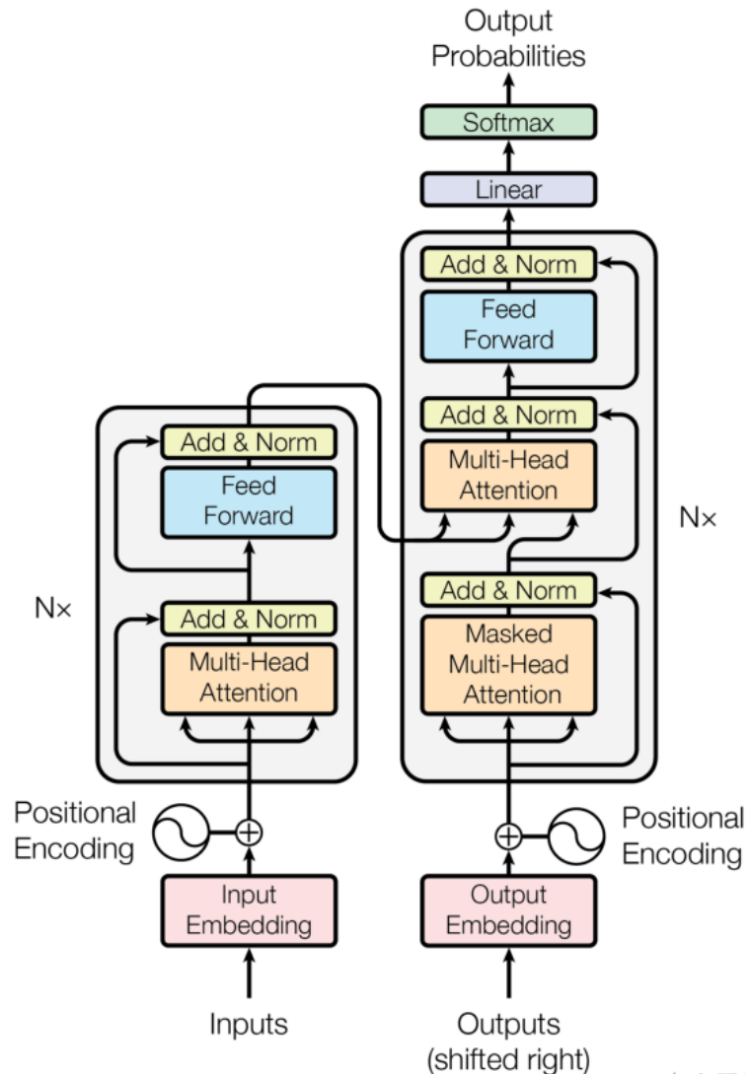


Рис. 6: Архитектурная схема Transformer.

параметров матрицы. d_k - размерность вектора Key. Когда размерность двух векторов велика, дисперсия результатов их точечного произведения также будет относительно велика, и градиент в этом случае может исчезнуть. Для того чтобы градиент оставался стабильным во время обучения, его необходимо разделить на d_k . По сравнению с нейронными сетями LSTM, Transformer более способен моделировать длинные последовательности. Кроме того, Transformer может использовать механизм самовнушения для достижения параллельных вычислений, что значительно повышает скорость обучения модели.

$$Att(q, k, v) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Музыкальная композиция в значительной степени опирается на повторяющиеся техники, такие как структура АВВС (А, В, С как музыкальные отрывки), и поскольку transformer, основанный на механизме самовнимания, эффективен в захвате информации на расстоянии от входной последовательности, он достиг впечатляющих результатов в других генеративных задачах, требующих длительной структурной согласованности. Поэтому применение Transformer к задачам моделирования и генерации музыки является многообещающим, и в настоящее время Transformer является наиболее широко изученным в области генерации символьной музыки.

1.4.3. Модель GAN

Вдохновленные теорией игр, генеративные состязательные сети состоят из двух подсетей - генератора и дискриминатора. Как показано на рисунке ниже 7. Генератор используется для генерации данных, близких к входным, а дискриминатор - для отличия реальных данных от сгенерированных. Обе сети должны постоянно оптимизировать свои генеративные и дискриминационные способности для достижения равновесия по Нэшу путем непрерывного чередующегося обучения состязательности. Было показано, что GAN и его разновидности генерируют генеративные сети, которые очень похожи на входные данные в таких областях, как генерация изображений и обработка естественного языка [16]. Функция потерь GAN показана в уравнении.

$$\min_G \max_D \mathbb{E}_x \sim p_{data} [\log D(x)] + \mathbb{E}_Z \sim p_z [\log(1 - D(G(Z)))]$$

В уравнении D обозначает генератор, G обозначает дискриминатор, x обозначает реальные данные, p_{data} обозначает плотность распределения вероятности реальных данных, а z обозначает случайные входные данные.

В задачах генерации символьной музыки GAN часто комбинируется с другими глубокими нейронными сетями или методами для генерации музыкальных последовательностей более высокого качества.

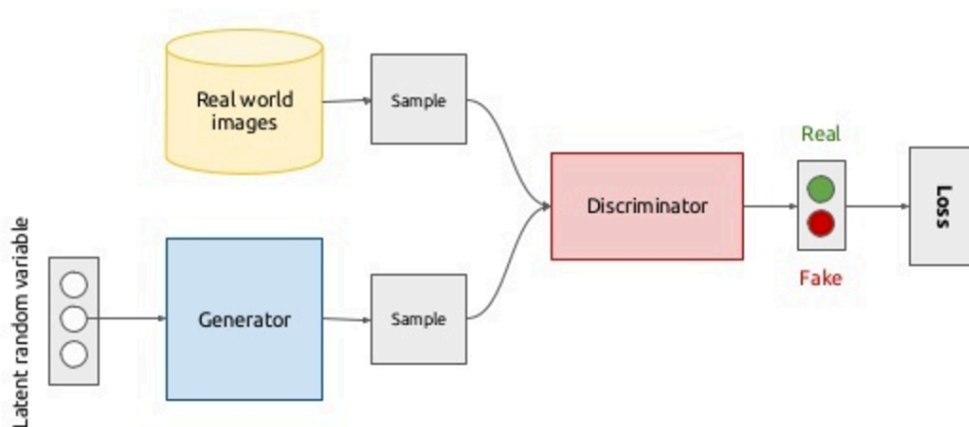


Рис. 7: Архитектурная схема GAN.

2. Классическая и усовершенствование модель

В этом разделе для сравнительных экспериментов используются три модели из разных периодов исследований в области генерации символической музыки на основе глубокого обучения, каждая из которых использует различные сетевые архитектуры и методы характеристики, чтобы дать более интуитивное представление о последних разработках в области генерации символической музыки.

Конфигурация сервера, используемая для экспериментов, представляет собой процессор AMD Ryzen9 5900HX с 32 ГБ оперативной памяти и GPU RTX 3080 с 12 ГБ видеопамати. для простоты измерений для обучения используется один и тот же набор данных - midi-файл из 100 полифонических музыкальных треков.

Python был выбран в качестве языка программирования для этого эксперимента из-за его превосходства над другими языками для задач машинного обучения и поддержки большого количества полезных библиотек. Для обработки музыкальных данных было решено выбрать библиотеку Music21, поскольку она отлично справляется с MIDI-файлами. В качестве программы визуализации MIDI мы использовали musescore.

В задаче создания символической музыки текущие основные генеративные модели в основном используют LSTM, Transformer и GAN в

качестве базовой архитектуры.

2.1. Классические модели

2.1.1. Генерация музыки на LSTM и Transformer

Для изучения простой задачи генерации фортепианной музыки мы обучили обе задачи генерации на одном и том же наборе данных. Для фортепианной музыки наиболее классическими являются только упражнения Баха, поэтому здесь мы выбрали 36 миди-файлов Баха. Хотя набор данных небольшой, он очень классический. В итоге мы сгенерировали один трек фортепианной музыки. На рисунке 8 показан XML-файл небольшого миди-файла Баха.



Рис. 8: Музыкальная нотация.

Для лучшего качества сравнения обе модели имеют одинаковую архитектуру, как показано ниже. Здесь мы генерируем музыкальный отрывок, используя ту же итерационную технику, которую мы использовали для генерирования текстовых последовательностей, следующим образом:

1. Учитывая текущую последовательность (нот и длительностей), модель предсказывает два распределения, одно для следующей ноты и одно для следующей длительности.

2. Мы делаем выборку из обоих этих распределений, используя параметр температуры, чтобы контролировать, насколько сильно мы хотим варьировать процесс выборки.
3. Выбранные нота и длительность добавляются к соответствующим входным последовательностям.
4. Процесс повторяется с новыми входными последовательностями для такого количества элементов, которое мы хотим сгенерировать.

Архитектура моделей на основе LSTM и трансформатора показана ниже на рисунке 9.

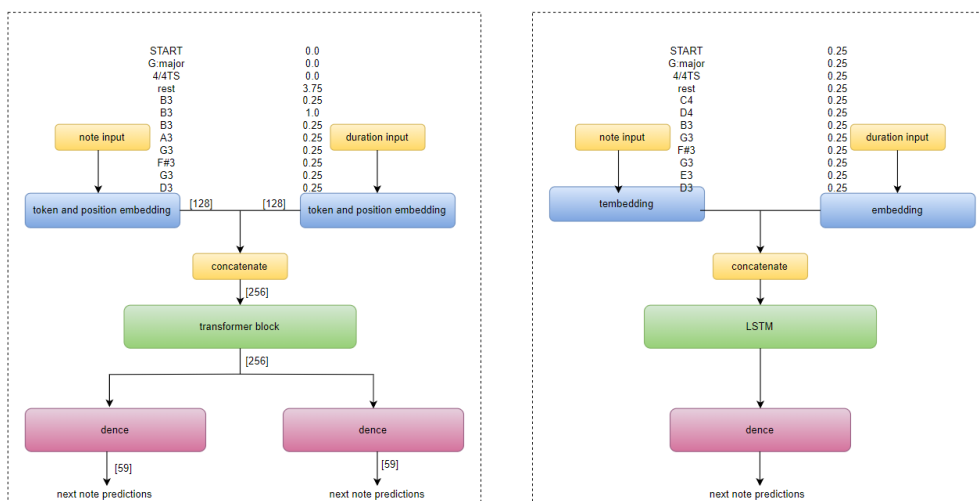


Рис. 9: Архитектура модели LSTM и Transformer.

Успех подхода к созданию музыки на основе Transformer часто сильно зависит от выбора метода токенизации. Здесь показано, как два вкрапления ("токен" и "позиция") складываются для получения общего вкрапления для последовательности, показана ниже на рисунке 10.

Далее мы анализируем результаты эксперимента: во-первых, по времени обучения модель LSTM намного быстрее, чем модель Transformer. А для сгенерированных музыкальных клипов, как показано на рисунке 11. При значении Epoch=50 сгенерированный трансформатором фрагмент имеет более музыкальное и ритмичное звучание и более характерен для упражнения; при значении Epoch=100 сгенерированный транс-

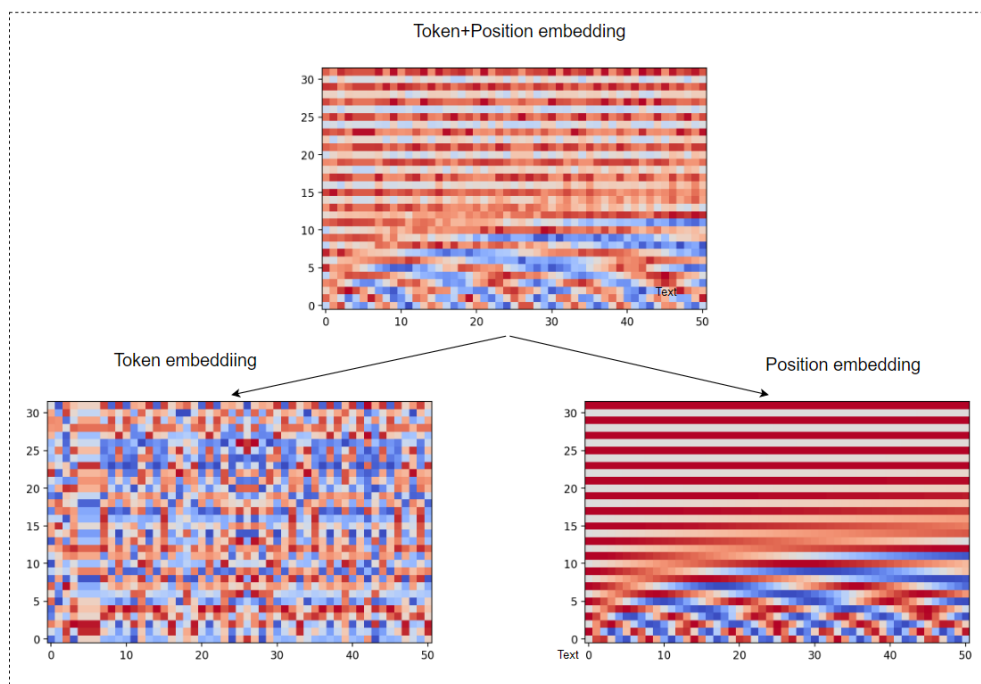


Рис. 10: Встраивание токенов и позиций.

форматором фрагмент представляет собой произведение в размере $3/4$ с LSTM на до-мажоре имеет восходящий и нисходящий знак, что неприятно слышать, и в нем присутствует нота более чем на одну октаву, что влияет на эффект. Наконец, при Epoch=150 в LSTM есть близкие двойные тона и их много, что может вызвать диссонанс, а в Transformer также есть тон, который начинается с пробелом более двух тактов, что может вызвать отсутствие ритма в общем фрагменте. В целом, сгенерированные образцы неплохи.

Что касается генерируемых последовательностей, Transformer генерирует несколько более богатые последовательности по сравнению с LSTM. Как показано на рисунке ниже на рисунке 12.

LSTM - это классическая рекуррентная нейронная сеть, которая может быть использована для обработки последовательных данных и поэтому хорошо подходит для генерации музыкальных последовательностей. LSTM относительно быстро обучается, поскольку ей нужно обрабатывать только скрытое состояние предыдущего момента и текущий вход. LSTM может генерировать последовательности последовательных нот и поэтому может генерировать музыку в течение длительных



Рис. 11: Сгенерированные музыкальные дорожки.

периодов времени. Однако модель LSTM имеет ограниченный объем памяти, чтобы уловить долгосрочные зависимости. LSTM склонна к чрезмерной подгонке, так как запоминает детали и шум обучающих данных. Музыка, генерируемая LSTM, может показывать повторяющиеся паттерны, так как ей не хватает информации о глобальном контексте. Transformer - это современная нейросетевая модель, которая хорошо работает с последовательными данными и особенно подходит для длинных последовательностей. Модели transformer обладают сильной способностью к запоминанию и могут улавливать долгосрочные зависимости. Модели transformer могут изучать особенности последовательностей на глобальном уровне и поэтому могут генерировать более сложную и разнообразную музыку. Трансформаторная модель относительно медленно обучается, поскольку ей необходимо обработать информацию обо всей последовательности. сложность трансформаторной модели высока и требует больше вычислительных ресурсов и обучающих данных. музыка, генерируемая трансформатором, может страдать от разрывов, поскольку она основана на самовнимательном механизме

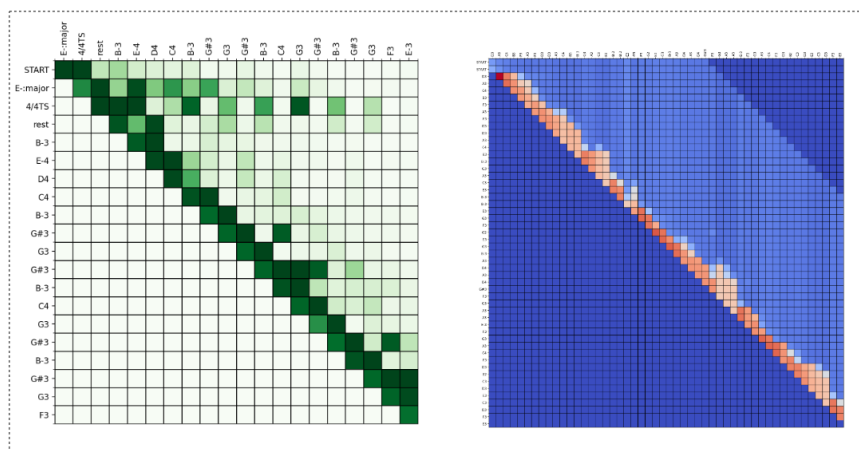


Рис. 12: Генерация последовательностей Модели Transformer и LSTM.

генерирования музыки и может иметь некоторые несоответствия. Музыка, генерируемая Трансформером, может страдать прерывистостью, поскольку она основана на механизме самовнушения и может иметь некоторые бессвязные ноты и аккорды.

2.1.2. Генерация музыки на MuseGAN

Здесь мы можем применить те же методы на основе конволюции, которые так хорошо работают для решения задач генерации изображений - в частности, GANs. MuseGAN был представлен в работе 2017 года "MuseGAN: Multi-Track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment". Авторы показывают, как можно обучить модель для генерации полифонической, многодорожечной, многобарной музыки с помощью новой структуры GAN. Более того, они показывают, как, разделив ответственность за векторы шума, которые питают генератор, они могут поддерживать тонкий контроль над высокоуровневыми временными и трековыми характеристиками музыки.

MuseGAN использует свертки для генерации полифонических музыкальных партитур с несколькими дорожками, рассматривая партитуру как своего рода изображение, где дорожки являются отдельными каналами изображения. Новизна MuseGAN заключается в том, как четыре входных вектора шума (аккорды, стиль, мелодия и грув) орга-

низованы таким образом, что можно сохранить полный контроль над высокоуровневыми характеристиками музыки. Хотя базовая гармонизация все еще не столь совершенна или разнообразна, как у Баха, это хорошая попытка решить чрезвычайно сложную задачу, и она подчеркивает возможности GAN для решения широкого спектра проблем.

Мы видим на рисунке 13, есть четыре различных входа, которые питают генератор: Аккорды, стиль, мелодия и грув.

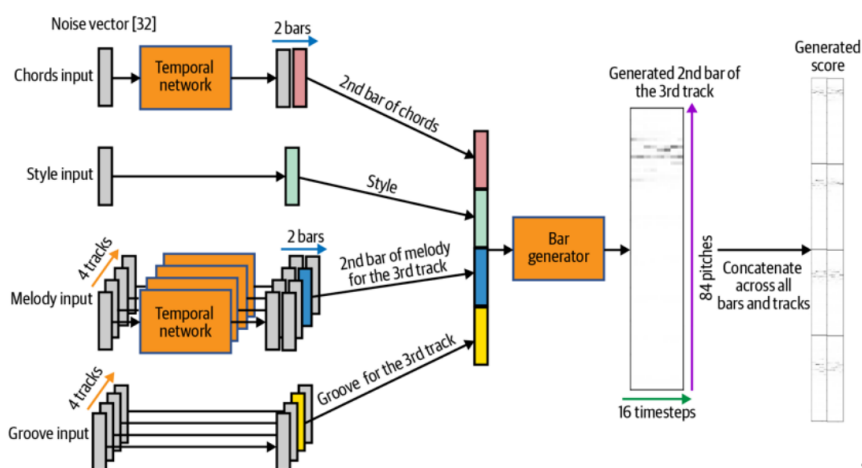


Рис. 13: Musegan generator.

Как и для дискриминатора, входными данными для критика является массив многодорожечных, многобарных оценок, каждая из которых имеет форму $[N_BARS, N_STEPS_PER_BAR, N_PITCHES, N_TRACKS]$.

Сначала мы свернем тензор вдоль оси брусьев. Мы применяем слои Conv3D по всей критике, так как работаем с четырехмерными тензорами. Затем мы свернем тензор вдоль оси шага. Наконец, мы свернем тензор по оси временных интервалов. На выходе получается плотный слой с одним блоком и без функции активации.

После 6 000 итераций мы получили окончательный сгенерированный образец двух тактов полифонической музыки с четырьмя дорожками. Из сгенерированных образцов видно, что с увеличением числа итераций сгенерированные образцы становятся более слышимыми, а аккорды - более гармоничными.

Объективная оценка модели представлена на следующем рисунке,

где показана диаграмма потерь для данной функции.

Субъективная оценка модели частично получена из анкеты и представлена на рисунке 13.

2.2. Классическая модель и ее усовершенствование

Здесь мы выбрали только LSTM и GAN, которые являются частью большой генеративной задачи, основанной на больших массивах данных, обученных генерировать полное однопорожечное музыкальное произведение.

2.2.1. Усовершенствованные модели LSTM

В данном эксперименте реализовано программное обеспечение для автоматической генерации музыки на основе машинного обучения. Основная функция программы - обучение LSTM-модели для генерации короткой песни и ее воспроизведения. Большинство современных генераторов музыки основаны на простых реализациях RNN и WaveNet, однако, из-за ограничений этих моделей, музыка, сгенерированная с помощью этих двух моделей, сильно гомогенизирована и плохо прослушивается. Это происходит потому, что традиционные нейронные сети не в состоянии сделать это на всей сцене. Для того чтобы улучшить качество генерируемой музыки, в основу программного обеспечения была положена модель LSTM. Мы ожидаем, что эта модель улучшит когерентность генерируемой музыки, чтобы повысить ее качество.

Программное обеспечение разделено на три основных модуля, как показано на рисунке 14, модуль генерации музыки, модуль воспроизведения аудио и пользовательский интерфейс (UI), как показано на рисунке 15. модуль генерации музыки генерирует музыку длиной около двух минут на основе входных данных, детали которых показаны на рисунке 15, а модуль воспроизведения аудио воспроизводит аудиофайл, введя его адрес, позволяя пользователю делать паузы, регулировать громкость и просматривать ход воспроизведения. пользовательский интерфейс обеспечивает функции программного обеспечения Интерфейс

с ПОЛЬЗОВАТЕЛЕМ.

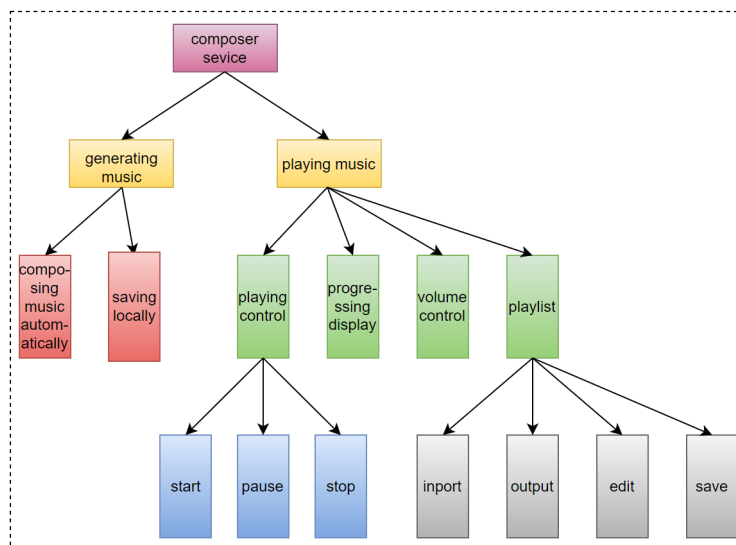


Рис. 14: Структурная схема программной системы.

Обучающая часть LSTM-модели для генератора показана ниже:

1. Получить все MIDI-файлы из каталога 'midi_songs', в данном случае набор midi данных 92 музыкальных произведений классического композитора-пианиста.
2. Используйте модуль 'music21' для разбора MIDI-файлов на ноты и аккорды.
3. Преобразовать ноты и аккорды в цифровую последовательность и разделить их на входную и выходную последовательность.
4. Создать нейронную сеть с тремя слоями LSTM и обучить ее с помощью оптимизатора 'rmsprop'.
5. Сохранить оптимальные веса для каждой эпохи в файл в процессе обучения.
6. Вызвать функцию 'train_network()' для выполнения процесса обучения.

Архитектура модели LSTM выглядит следующим образом:

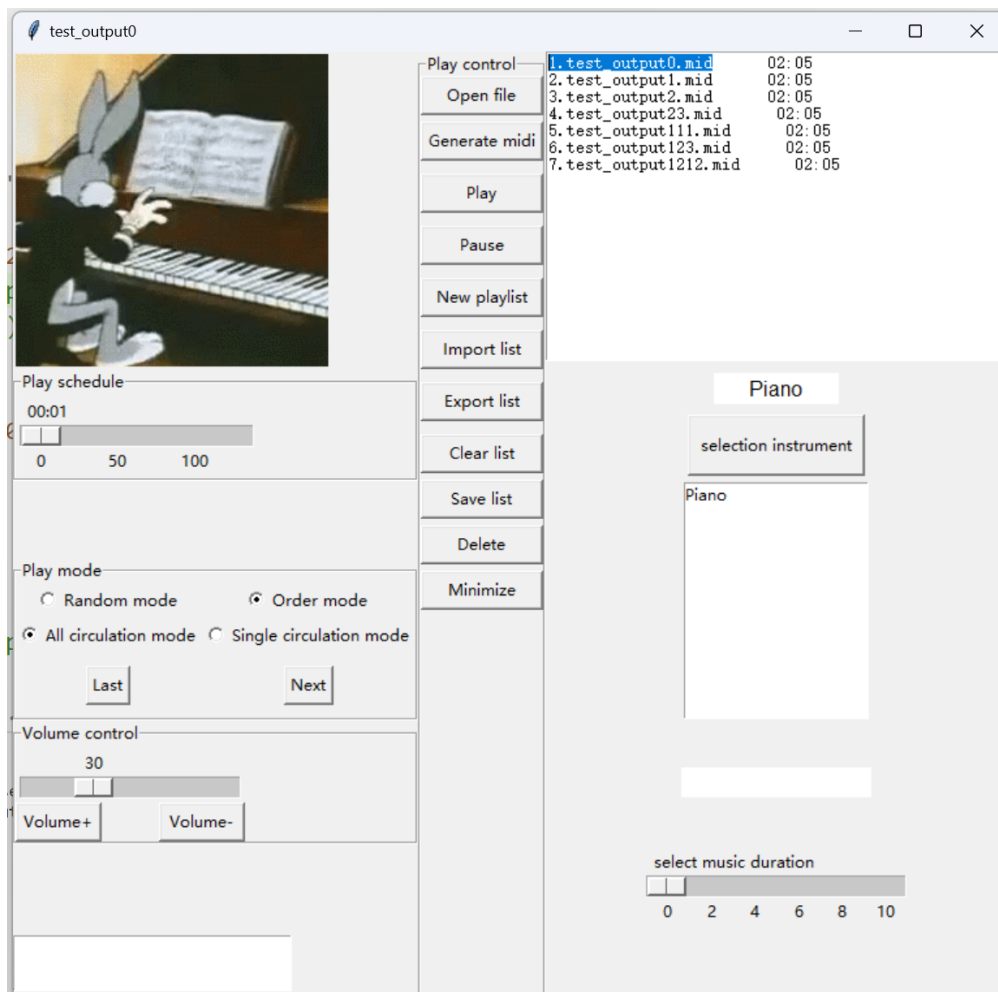


Рис. 15: Пользовательский интерфейс.

1. Слой LSTM, 512 ячеек, входная форма '(sequence_length, 1)', использование параметра 'recurrent_dropout=0.3', возврат последовательности
2. Слой LSTM, 512 ячеек, использование параметра 'recurrent_dropout=0.3', возвращение последовательности
3. Слой LSTM, 512 ячеек, последовательность не возвращается
4. Слой BatchNormalization
5. Слой Dropout с коэффициентом выпадения 0.3
6. Полностью связанный слой, 256 ячеек, использование функции активации ReLU

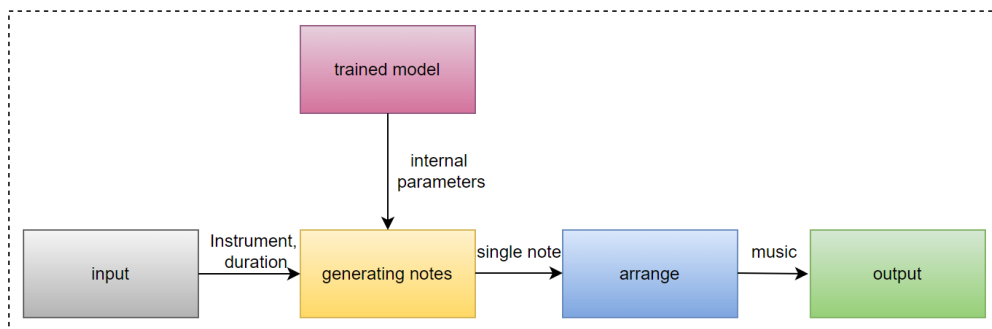


Рис. 16: Модуль генерации музыки.

7. Слой пакетной нормализации

8. Выпадающий слой с коэффициентом выпадения 0,3

9. Полностью связанный слой с выходной размерностью n_vocab , с использованием функции активации `softmax`.

Генерируемый музыкальный клип показан на рисунке 17 ниже, музыкальное произведение продолжительностью около двух минут.



Рис. 17: Сгенерированные музыкальные образцы с помощью MusicScore.

2.2.2. Усовершенствованные модели GAN

Мы выбрали Баха в качестве репрезентативного композитора для демонстрации работы по обучению.

Для решения задачи генерации на основе GAN мы выбрали большой набор данных фортепианных midi, состоящий из музыкальных композиций более чем двадцати музыкантов, который был загружен с музыкального плеера в Китае. Поскольку на большинстве музыкальных сайтов преобладает музыкальный жанр Mp3, мы предоставили коды преобразования Mp3 to wave и wave to midi для преобразования всех файлов в файлы midi. Мы провели анализ данных на этом наборе данных, чтобы показать двадцать музыкантов с наибольшим количеством произведений, имеющих чрезвычайно высокую долю произведений, как показано на рисунке 18, и представить график доли времени каждого автора, как показано на рисунке 19.

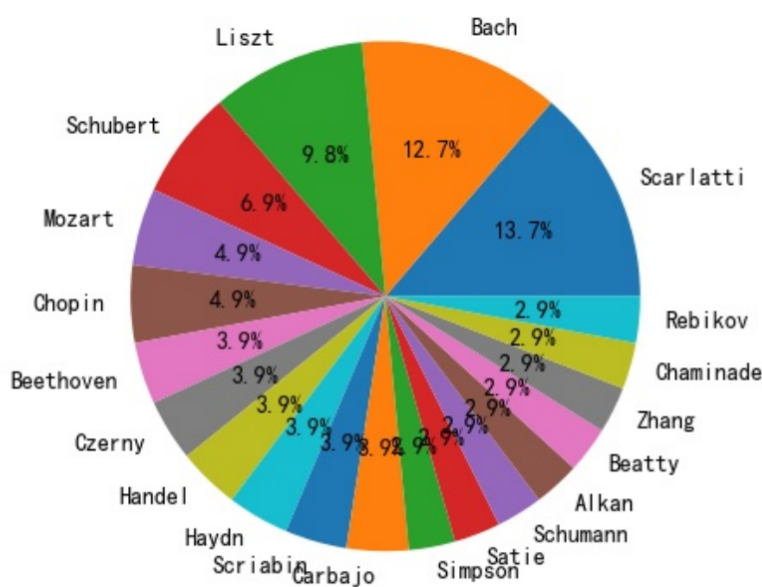


Рис. 18: Доля произведений композиторов.

Цель этого эксперимента - генерировать музыку в стиле каждого композитора, поэтому, хотя обучающие модели одинаковы, используемые наборы данных - это их собственная музыка. Мы подготовили список композиций для каждого композитора и используем его в качестве набора данных midi для обучения моделей.

Мы выбрали Баха в качестве репрезентативного композитора для демонстрации работы по обучению. На рисунке 20 показана гистограм-

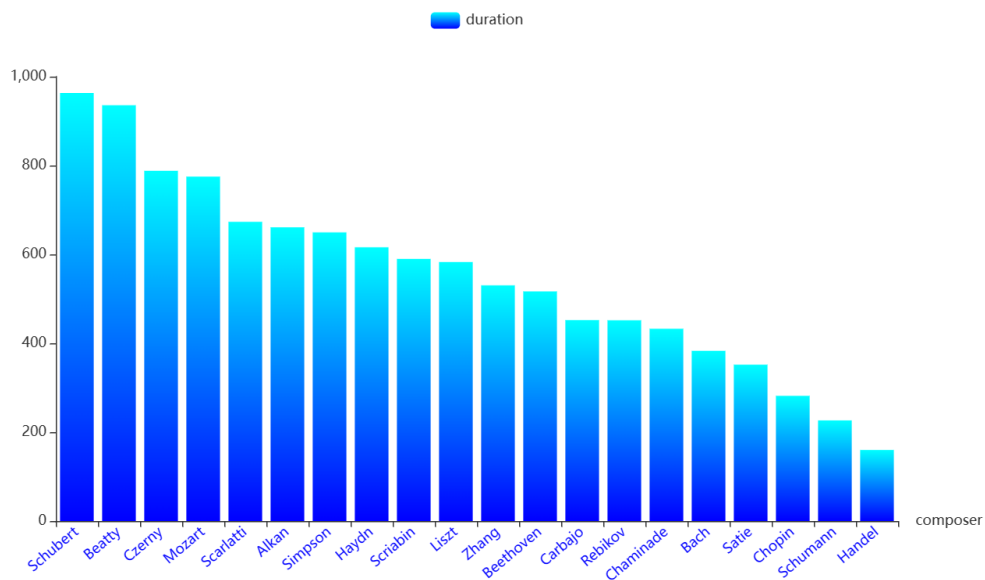


Рис. 19: Средняя продолжительность композиторских произведений.

ма частот нот Баха.

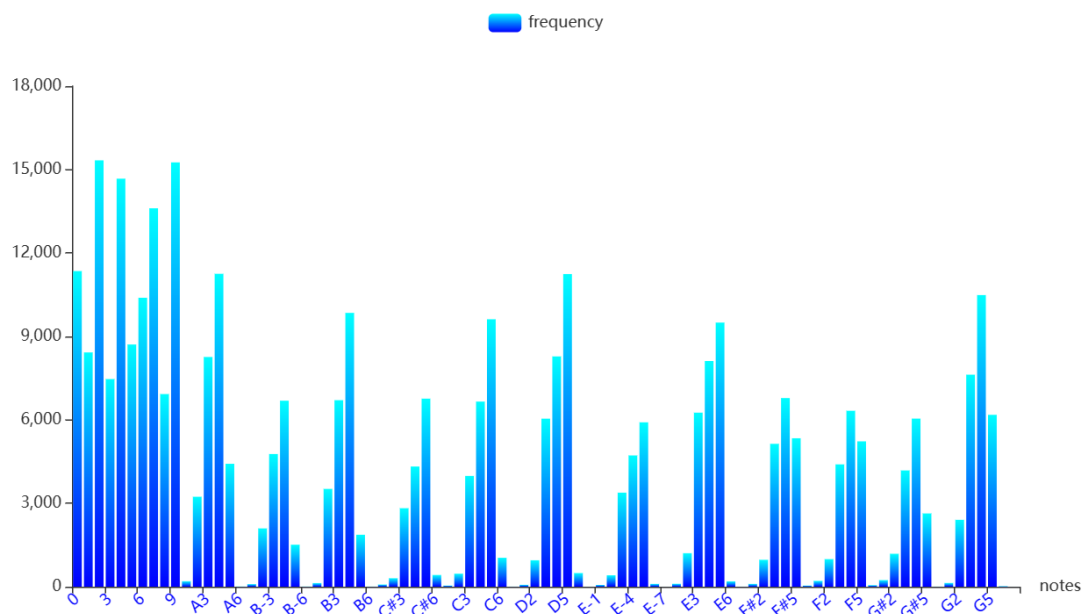


Рис. 20: Нотно-частотная диаграмма сочинений Баха.

Идея конструкции этого генератора показана на следующей схеме:

1. Поместить список музыки в список
2. Инициализация параметров
3. Создать дискриминатор, модель, как показано на рисунке 21.

4. Создать генератор, как показано на рисунке 22.
5. Начать обучение: Сначала используйте `get_notes`: Получите все ноты и аккорды из `mid` файлов. Затем используйте `prepare_sequences`:

```
model.add(Conv1D(filters=16, kernel_size=1, activation='relu'))
model.add(Conv1D(filters=16, kernel_size=1, activation='relu'))
model.add(MaxPooling1D(pool_size=1))
model.add(Flatten())
model.add(RepeatVector(1))
model.add(LSTM(10, activation='relu', return_sequences=False))
model.add(Dense(1, activation='sigmoid'))
```

Рис. 21: Модель Дискриминатор(python code).

```
model.add(Dense(16, input_dim=self.latent_dim))
model.add(LeakyReLU(alpha=0.2))
model.add(BatchNormalization(momentum=0.8))
model.add(Dense(32))
model.add(LeakyReLU(alpha=0.2))
model.add(BatchNormalization(momentum=0.8))
model.add(Dense(64))
model.add(LeakyReLU(alpha=0.2))
model.add(BatchNormalization(momentum=0.8))
model.add(Dense(np.prod(self.seq_shape), activation='tanh'))
model.add(Reshape(self.seq_shape))
```

Рис. 22: Модель Генератор(python code).

- Подготовьте последовательности,
 - Используемые нейронной сетью. Ввод в тренер для обучения.
6. Сгенерируйте и нарисуйте график: `self.generate(notes, epoch)` Генерирование музыки с помощью обученного генератора. Наконец, `plot_loss` построить график потерь, как показано на рисунке 23.

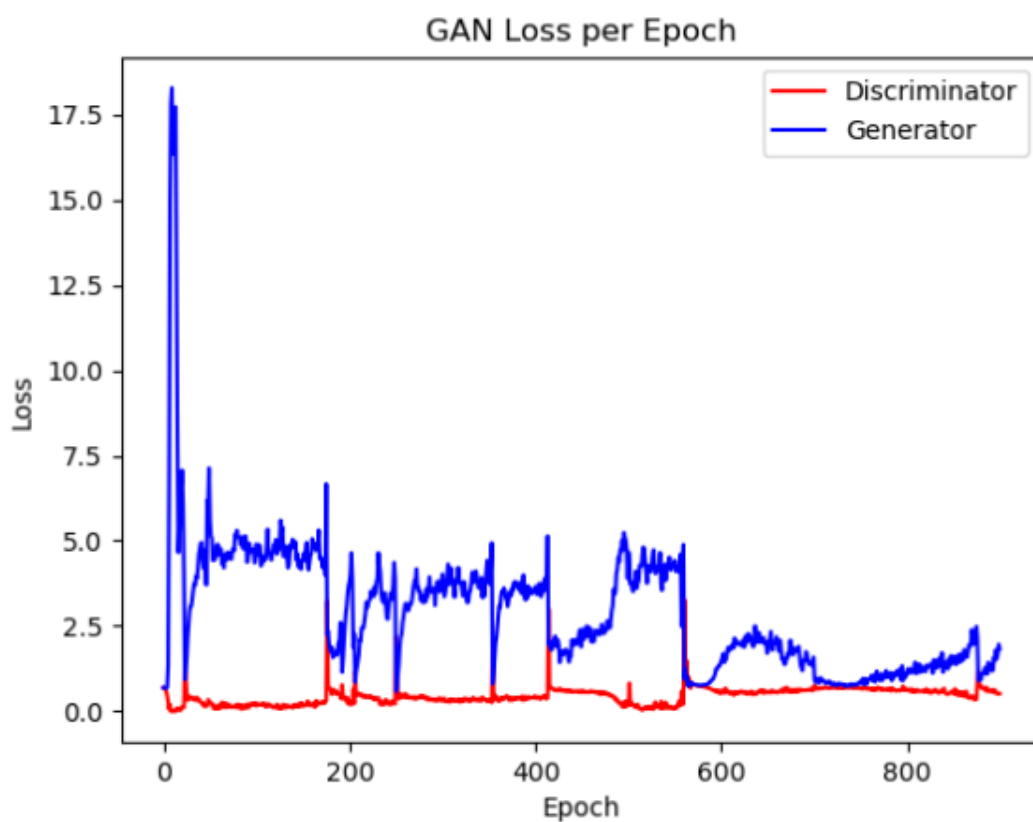


Рис. 23: График потер.

2.3. Субъективная оценка результатов

В этом разделе мы представляем исследование и результаты субъективной оценки музыкальных дорожек, сгенерированных нейронной сетью.

Мы поместили восемь образцов генерации в виде онлайн-анкеты, по два образца из экспериментов по сравнению LSTM и трансформатора, два образца генерации из MuseGAN, один образец из трехслойного генератора LSTM и образец из генератора GAN.

Во-первых, мы разделили участников опроса на четыре категории: работники музыки, связанные с музыкой, люди с некоторым музыкальным образованием, любители музыки и обычные люди. Те, кто участвовал в опросе, показаны на рисунке 24. Всего в опросе приняли участие 46 человек, в том числе работники музыки: 2, связанные с музыкой: 11, люди с некоторым музыкальным образованием: 32, любители музыки и обычные люди: 1.

What level of musical proficiency do you have?
(46 条回复)

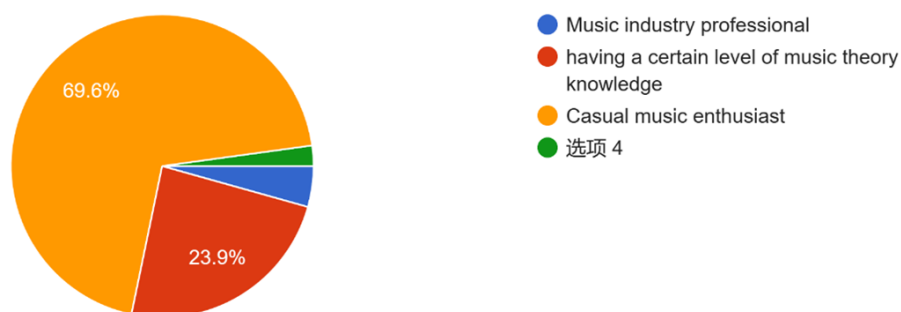


Рис. 24: Результаты опроса.

Для LSTM и Transformer результаты генерации двух музыкальных треков показаны на рисунке 25. В первом наборе данных мы получаем одинаковые оценки для обеих моделей, по 23 голоса в каждой группе. А во втором наборе данных мы получили более высокие субъективные оценки для Transformer, чем для LSTM, 29 человек предпочли музыку, созданную Transformer, и 17 человек предпочли музыку, созданную LSTM, причем Transformer получил на 12 голосов больше, чем

LSTM. Видно, что люди больше предпочитают сгенерированные образцы Трансформера, чем образцы LSTM.

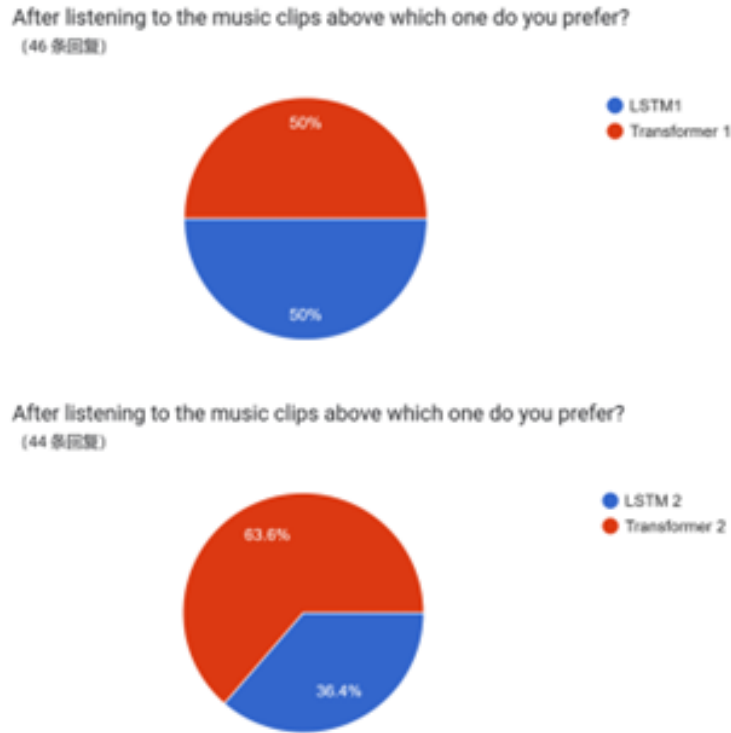


Рис. 25: Результаты опроса LSTV&Transformer.

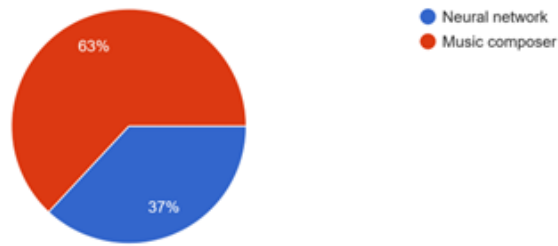
Для модели полифонической музыки MuseGAN результаты показаны на рисунке 26, как видно, сгенерированные образцы можно принять за следы работы человека. Это говорит о том, что образцы хорошо звучат и что модель MuseGAN генерирует хороший дорожек.

Для усовершенствованных моделей LSTM и GAN мы оценивали сгенерированные музыкальные клипы по четырем параметрам: гармония, плавность, полнота и художественная эстетика музыки. Для сгенерированной оценки для LSTM результат показан на рисунке 27. Видно, что для LSTM оценка 4 за гармонию, 3 за гладкость, 4 за полноту и 3 за художественную эстетику дает суммарную оценку 3,5, что является хорошим образцом.

Для сгенерированной оценки для трехслойной GAN результат показан на рисунке 28. Видно, что для GAN оценка 1 за гармонию, 2 и 5 за гладкость, усредненная для получения оценки 3,5, 4 за полноту и 5 за художественную эстетику, и общая оценка 3,375 привела к плохому

Who do you think "Musegan1" was composed by?

(46 条回复)



Who do you think "Musegan4" was composed by?

(46 条回复)

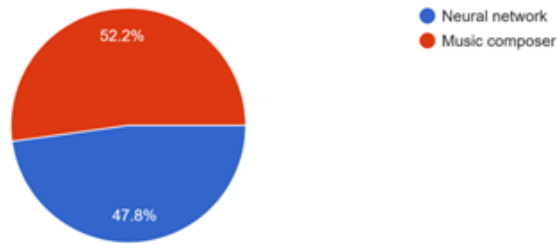


Рис. 26: Результаты опроса MuseGAN.

рейтингу для этого образца из-за плохой гармонии.

Таким образом, в целом, исследование показало, что лучшей выборкой была: Классические модели: MuseGAN, Transformer и LSTM; Усовершенствованные модели: LSTM, затем GAN.

Ссылка на анкету: Субъективная оценка для генерируемых нейронной сетью музыкальных дорожек <https://forms.gle/b33RUH5JGWKktVVGa>.

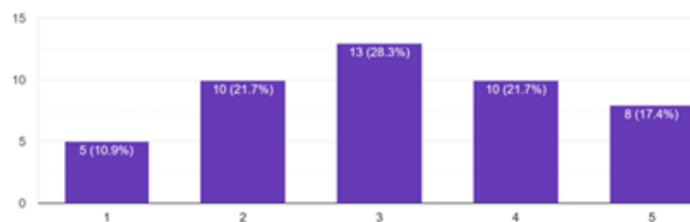
From the completeness of the music

(46 条回复)



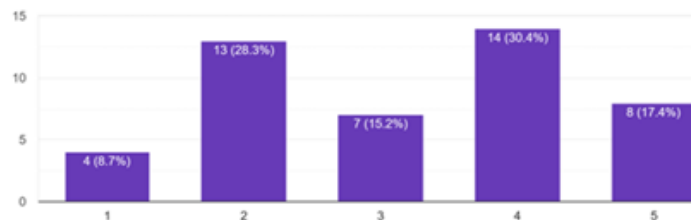
From the smoothness of the music

(46 条回复)



From the harmony of the music

(46 条回复)



From the artistic aesthetic of the music

(46 条回复)

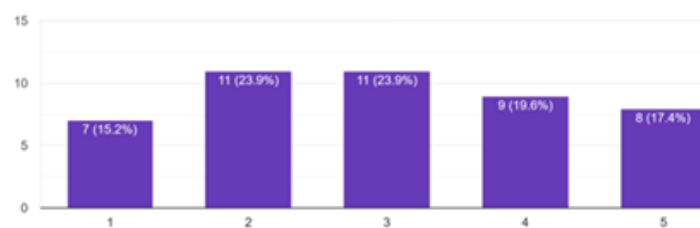
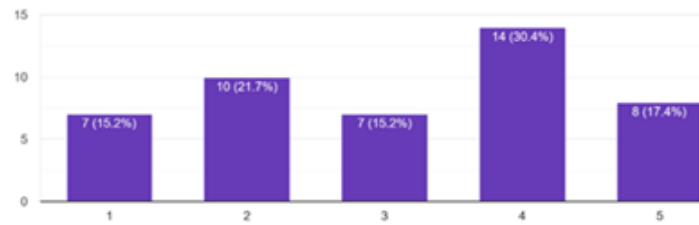


Рис. 27: Результаты опроса 3-layer-LSTM.

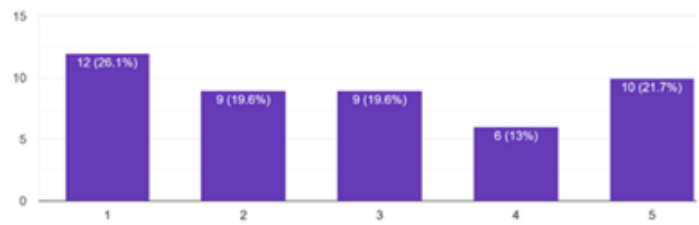
From the completeness of the music
(46 个回答)



From the smoothness of the music
(46 个回答)



From the harmony of the music
(46 个回答)



From the artistic aesthetic of the music
(46 个回答)



Рис. 28: Результаты опроса GAN.

3. Заключение

Целью работы является исследование методов генерации фортепианной музыки с использованием нейронных сетей, таких как LSTM, Transformer и GAN, обучение нескольких моделей генерации фортепианной музыки и оценка их производительности. Кроме того, в данной работе оценивается субъективное качество сгенерированных музыкальных клипов с помощью анкеты для выбора лучшей модели генерации фортепианной музыки, а также разрабатывается и внедряется приложение для автоматической генерации музыки на основе этой модели.

Основные задачи данной работы заключаются в следующем:

Во-первых, мы собрали и подготовили данные, используемые для проведения экспериментов, выбрали три классические модели LSTM, GAN и Transformer, сгенерировали с помощью этих моделей односторонние и многосторонние фрагменты фортепианной музыки, а также выбрали усовершенствованные модели LSTM и GAN для обучения генерированию фортепианной музыки длительностью около 2 мин.

Во-вторых, в данной работе объективно оцениваются три модели генерации музыки на основе нейронных сетей и субъективно оценивается качество сгенерированных музыкальных фрагментов для выбора лучшей модели генерации фортепианной музыкальной нотации, что доказывает эффективность и мастерство предложенного метода генерации фортепианной музыкальной нотации. Среди них, для анкетной части, мы выбрали музыкальные клипы, сгенерированные каждой из этих моделей, два клипа классической LSTM и Transformer с хорошим опытом прослушивания, два клипа GAN и по одному клипу продвинутой LSTM и GAN, чтобы провести анкетный опрос, сформировать результаты анкетирования и проанализировать их.

Наконец, в данной работе разрабатывается и внедряется приложение для автоматической генерации музыки на основе лучшей модели генерации музыки фортепианной нотации. В части разработки программного обеспечения мы выбираем лучшую модель среди субъективных моделей и разрабатываем программное обеспечение для авто-

матической генерации музыки на основе машинного обучения LSTM-composer, предоставляя структуру и функции программной системы. Это может дать создателям и энтузиастам музыки больше вдохновения и творческих инструментов, имеет широкие перспективы применения и коммерческую ценность.

Таким образом, данное исследование имеет важное теоретическое и практическое значение для исследования и применения технологии генерации фортепианной музыки, а также предоставляет новые идеи и методы для развития технологии искусственного интеллекта в музыкальной индустрии.

Исходный код опубликован на платформе GitHub по ссылке <https://github.com/wumetay/neural-networks-generate-music-tracks>.

Список литературы

- [1] Vaswani Ashish, Shazeer Noam M., Parmar Niki, Uszkoreit Jakob, Jones Llion, Gomez Aidan N., Kaiser Lukasz, and Polosukhin Illia. Attention is All you Need // NIPS. — 2017.
- [2] Baggi Denis and Haus Goffredo. IEEE 1599: Music encoding and interaction // Computer. — 2009. — 04. — Vol. 42. — P. 84 – 87.
- [3] Boulanger-Lewandowski Nicolas, Bengio Y., and Vincent Pascal. Modeling Temporal Dependencies in High-Dimensional Sequences: Application to Polyphonic Music Generation and Transcription // Proceedings of the 29th International Conference on Machine Learning, ICML 2012. — 2012. — 06. — Vol. 2.
- [4] Bretan Mason, Weinberg Gil, and Heck Larry. A Unit Selection Methodology for Music Generation Using Deep Neural Networks. — 2016. — 12.
- [5] Briot Jean-Pierre and Pachet Francois. Music Generation by Deep Learning - Challenges and Directions // Neural Computing and Applications. — 2020. — 02. — Vol. 32.
- [6] Yang Wei, Sun Ping, Zhang Yi, and Zhang Ying. CLSTMS: A Combination of Two LSTM Models to Generate Chords Accompaniment for Symbolic Melody. — 2019. — 05. — P. 176–180.
- [7] Eck Douglas and Schmidhuber Juergen. A First Look at Music Composition using LSTM Recurrent Neural Networks. — 2002.
- [8] Chen K., Zhang Weilin, Dubnov Shlomo, and Xia Gus. The Effect of Explicit Structure Encoding of Deep Neural Networks for Symbolic Music Generation // 2019 International Workshop on Multilayer Music Representation and Processing (MMRP). — 2018. — P. 77–84.
- [9] Elman Jeffrey L. Finding Structure in Time // Cogn. Sci. — 1990. — Vol. 14. — P. 179–211.

- [10] Gatys Leon, Ecker Alexander, and Bethge Matthias. Image Style Transfer Using Convolutional Neural Networks. — 2016. — 06. — P. 2414–2423.
- [11] Wu Shangda, Yang Yue, Wang Zhaowen, Li Xiaobing, and Sun Maosong. Generating Chords from Melody with Flexible Harmonic Rhythm and Controllable Harmonic Density. — 2021.
- [12] Goodfellow Ian J., Pouget-Abadie Jean, Mirza Mehdi, Xu Bing, Warde-Farley David, Ozair Sherjil, Courville Aaron C., and Bengio Yoshua. Generative Adversarial Nets // NIPS. — 2014.
- [13] Goel Kratarth, Vohra Raunaq, and Sahoo Jajati. Polyphonic Music Generation by Modeling Temporal Dependencies Using a RNN-DBN. — 2014. — 12. — Vol. 8681. — P. 217–224.
- [14] Roberts Adam, Engel Jesse, Raffel Colin, Hawthorne Curtis, and Eck Douglas. A Hierarchical Latent Vector Model for Learning Long-Term Structure in Music. — 2018. — 03.
- [15] Hochreiter Sepp and Schmidhuber Jürgen. Long Short-Term Memory // Neural Computation. — 1997. — Vol. 9. — P. 1735–1780.
- [16] Huang He, Yu Phillip, and Wang Changhu. An Introduction to Image Synthesis with Generative Adversarial Nets. — 2018. — 03.
- [17] Hung Hsiao-Tzu, Wang Chung-Yang, Yang Yi-Hsuan, and Wang Hsin-Min. Improving Automatic Jazz Melody Generation by Transfer Learning Techniques // 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC). — 2019. — P. 339–346.
- [18] Kingma Diederik P. and Welling Max. Auto-Encoding Variational Bayes // CoRR. — 2013. — Vol. abs/1312.6114.
- [19] Donahue Chris, Mao Huanru Henry, Li Yiting, Cottrell G., and McAuley Julian. LakhNES: Improving Multi-instrumental Music Gen-

- eration with Cross-domain Pre-training // International Society for Music Information Retrieval Conference. — 2019.
- [20] Lim Hyungui, Rhyu Seungyeon, and Lee Kyogu. Chord Generation from Symbolic Melody Using BLSTM Networks. — 2017. — 10.
- [21] Lo Man Yat and Lucas Simon. Evolving Musical Sequences with N-Gram Based Trainable Fitness Functions. — 2006. — 01. — P. 601 – 608.
- [22] Brunner Gino, Konrad Andres, Wang Yuyi, and Wattenhofer Roger. MIDI-VAE: Modeling Dynamics and Instrumentation of Music with Applications to Style Transfer // ArXiv. — 2018. — Vol. abs/1809.07600.
- [23] Meredith David. Method of Computing the Pitch Names of Notes in MIDI-like Music Representations. — 2004. — 01.
- [24] Mozer Michael. Neural Network Music Composition by Prediction: Exploring the Benefits of Psychoacoustic Constraints and Multi-scale Processing // Connection Science - CONNECTION. — 1994. — 01. — Vol. 6. — P. 247–280.
- [25] Krishnan V., Subramanian Rajarajeswari, Krishnamohan Venkat, Sheel Vivek, and Rajendran Deepak. Music Generation Using Deep Learning Techniques // Journal of Computational and Theoretical Nanoscience. — 2020. — 07. — Vol. 17. — P. 3983–3987.
- [26] Wang Ziyu, Zhang Yiyi, Zhang Yixiao, Jiang Junyan, Yang Ruihan, Zhao Junbo Jake, and Xia Gus. PIANOTREE VAE: Structured Representation Learning for Polyphonic Music // International Society for Music Information Retrieval Conference. — 2020.
- [27] Pati Ashis, Lerch Alexander, and Hadjeres Gaëtan. Learning to Traverse Latent Spaces for Musical Score Inpainting. — 2019. — 11.
- [28] Pinkerton Richard C. Information theory and melody. // Scientific American. — 1956. — Vol. 194. — P. 77–87.

- [29] Lattner Stefan, Grachten Maarten, Agres Kat, and Cancino Chacón Carlos. Probabilistic Segmentation of Musical Sequences Using Restricted Boltzmann Machines. — 2015. — 06. — Vol. 9110.
- [30] Putnam Jeffrey. A Grammar-Based Genetic Programming Technique Applied to Music Generation. — 1996. — 01. — P. 363–368.
- [31] Sigtia Siddharth, Benetos Emmanouil, Cherla Srikanth, Weyde Tillman, d’Avila Garcez Artur S., and Dixon Simon. An RNN-based Music Language Model for Improving Automatic Music Transcription // International Society for Music Information Retrieval Conference. — 2014.
- [32] Schulze Walter and Van Der Merwe Brink. Music Generation with Markov Models // Multimedia, IEEE. — 2011. — 04. — Vol. 18. — P. 78 – 85.
- [33] Shao Xi, Xu Changsheng, and Kankanhalli Mohan. Unsupervised classification of music genre using hidden Markov model. — 2004. — 07. — P. 2023 – 2026 Vol.3.
- [34] Sheng Zhonghao, Song Kaitao, Tan Xu, Ren Yi, Ye Wei, Zhang Shikun, and Qin Tao. SongMASS: Automatic Song Writing with Pre-training and Alignment Constraint // ArXiv. — 2020. — Vol. abs/2012.05168.
- [35] Todd Peter. A Connectionist Approach to Algorithmic Composition // Computer Music Journal. — 1989. — 12. — Vol. 13.
- [36] Jiang Junyan, Xia Gus, Carlton Dave, Anderson Chris, and Miyakawa Ryan. Transformer VAE: A Hierarchical Model for Structure-Aware and Interpretable Music Representation Learning. — 2020. — 05. — P. 516–520.
- [37] Fang Yinyin, Xu Yong, Li Heju, He Xin, and Kang Longlong. Writing in the air: Recognize Letters Using Deep Learning Through WiFi Signals. — 2020. — 07. — P. 8–14.

- [38] Zhuang Chen and Jinming Yu. GCA:A chord music generation algorithm based on double-layer LSTM. — 2021. — 04. — P. 57–61.