

Содержание

Введение	4
1 Анализ предметной области	5
1.1 Актуальность темы	5
1.2 Виды музыкальных фрагментов	6
1.3 Проблемы при генерации музыки	6
1.3.1 Проблема креативности	7
1.3.2 Проблема глобальной структуры	7
1.3.3 Проблема интерактивности	7
1.4 Методы генерации музыки	8
1.4.1 Вероятностные методы	8
1.4.2 Грамматические методы	9
1.4.3 Методы искусственного интеллекта	9
1.4.4 Биологические методы	9
1.5 Вывод	10
2 Классификация существующих методов	11
2.1 Методы генерации музыкальных произведений полифонического склада	11
2.1.1 Марковские цепи	11
2.1.2 Контекстно-зависимая грамматика	12
2.1.3 Л-системы	13
2.1.4 Клеточные автоматы	14
2.1.5 Генетический алгоритм	14
2.1.6 Генеративно-состязательные сети	15
2.1.7 Рекуррентные нейронные сети	16
2.2 Сравнение и оценка методов	17
2.3 Вывод	18
Заключение	19
Список литературы	20

Введение

При появлении первых компьютерных машин, люди задумались о возможности автоматизировать написание картин и музыки. Поскольку сочинение музыки является долгим процессом, поскольку композитор проводит тяжелую работу, вопрос генерации музыки был крайне важен. Роль музыки в мире невозможно переоценить – практически в каждой игре, в каждом магазине, в каждом фильме используется какая-нибудь мелодия. Именно поэтому нам необходимо иметь возможность быстро сгенерировать музыкальный фрагмент, чтобы использовать его в нужном месте.

Целью данной работы является классификация современных методов генерации музыки полифонического склада. В рамках работы необходимо решить следующие задачи:

- рассмотреть виды генерируемых музыкальных фрагментов;
- выделить основные проблемы при генерации музыкального произведения;
- описать методы генерации музыки;
- провести сравнение методов на основании критерия оценки;
- выбрать алгоритм, который наиболее точен при решении поставленной задачи.

1 Анализ предметной области

В данном разделе будет рассмотрена актуальность темы, выбран объект генерации, а также рассмотрены проблемы генерации музыки и группы методов.

1.1 Актуальность темы

Вопрос автоматической генерации музыки является весьма важным в наше время. Данная технология находит широкое применение в различных областях.

Тема широко исследовалась, потому что считается, что все написание музыки можно алгоритмизировать. Поэтому было проведено множество экспериментов [1] с целью выявить закономерности в произведениях или разложить музыку на составляющие части. Все это нужно для того, чтобы понять – какой фрагмент музыкального произведения вызывает те или иные эмоции и почему. Ответы на эти вопросы позволяют написать эффективный алгоритм, который сможет правильно передать настроение, которое хочет получить пользователь и при этом очень быстро. Развитие данной темы привело к высокой потребности такого вида музыки.

Так, в сфере игр нередко саундтреки являются сгенерированы автоматически. В качестве примера можно привести специальную нейронную сеть под названием Neural Composer, которая была научена писать музыку на основе 4 тысяч различных композиций из видео игр [2]. Данный инструмент может помочь добавить в приложение музыкальное сопровождение, которое при этом будет уникальным. Подобное желание обуславливается несколькими факторами, которые легко решаются использованием автоматического генератора:

- невозможностью нанять музыканта для написания композиции;
- желанием получить уникальный, ранее не существовавший музыкальный фрагмент.

Применение автоматических генераторов является важной частью самой музыки. Существуют специальные приложения, такие как Guitar PRO или

Finale. Эти приложения позволяют автоматизировать процесс написания. Они обладают следующими возможностями:

- построить аккорд, имея заданную тональность;
- транспонировать произведение с одной тональности в другую;
- переписать музыкальное произведение на другой музыкальный инструмент.

1.2 Виды музыкальных фрагментов

При генерации произведения музыкальные фрагменты можно поделить на следующие виды [3].

1. Ритм или партия для ударных инструментов.
2. Аккомпанемент под мелодию из заранее заданного набора аккордов.
3. Вариация уже готовой мелодии.
4. Мелодия, где в один момент времени звучит не более одной ноты.
5. Полифоническая музыка, то есть музыка, где используется в один момент времени более одного инструмента.

Центральным видом в данной исследовательской работе является генерация полифонической музыки, поэтому речь далее идет именно об этом виде музыкального произведения.

1.3 Проблемы при генерации музыки

Поскольку написание музыки является глубоко творческим процессом, следовательно программа должна соответствовать данному требованию, из-за чего возникает ряд проблем.

1.3.1 Проблема креативности

Проблема креативности является центральной проблемой, которую необходимо решить при написании автоматического генератора. Дело заключается не только в художественной стороне вопроса, но и в экономической [4], поскольку при этом может возникнуть проблема с авторским правом. Генератор должен писать музыку, которая не похожа на ранее существующие мелодии, чтобы избежать данных проблем. Существует множество методов для решения этого вопроса, основанные на алгоритмах работы с текстом.

1.3.2 Проблема глобальной структуры

Произвольная комбинация нот может быть началом некоторой композиции. Продолжение первых комбинаций должно быть похоже на начало музыкальной композиции, иначе следующие комбинации будут похожи на начало нового произведения. Каждый последующий набор нот повторяет предыдущий или имеет некоторые изменения.

Подобные замечания накладывают на модель требования – уметь считать и копировать. Выполнить подобные требования необходимо для того, чтобы в итоге музыкальное произведение представляло из себя глобальную структуру. [4]. Однако при субъективном прослушивании результата ощутимо, что каждые 5-10 секунд произведение изменится, при этом оставаясь музыкой, что и называется проблемой глобальной структуры.

В качестве решения были предложены LSTM [5] сети, однако они не подходят для копирования более 16-20 фрагментов, когда для музыкальной композиции может понадобиться 256 и более.

1.3.3 Проблема интерактивности

В соответствии с работой Роберта Роу [4], интерактивная компьютерная музыкальная система - система, поведение которой меняется в ответ на музыкальный ввод.

В большинстве систем генерация автоматизирована, при этом с небольшим количеством интерактивности или совсем без нее. Из-за этого локальная модификация получившегося музыкального фрагмента обычно не поддерживается, из-за чего нужно делать полную регенерацию музыкального отрезка. Но суть заключается в том, что это не похоже на работу музыканта, который может изменить свое произведение в любой момент.

1.4 Методы генерации музыки

Можно выделить четыре основные группы методов, которые способны решить задачу генерации:

- вероятностные;
- грамматические;
- методы искусственного интеллекта;
- биологические.

1.4.1 Вероятностные методы

Вероятностные методы имеют широкое распространение. Случайность уже давно используется для генерации музыки. Большинство методов, однако, сильно ограничены тем, что они полагаются на составителя, который уже создал возможные такты музыки, которые нужно собрать вместе. Другими словами, пользователь просто по-новому собирает воедино уже сочиненную музыку [6]. При этом остается открытым следующий вопрос – как же случайным образом создавать отдельные ноты, которые составляют такты. Вероятностные методы занимаются решением подобных вопросов, используя саму теорию вероятности, а также цепи Маркова.

1.4.2 Грамматические методы

Грамматические методы генерации музыки основаны на использовании грамматик, а также алгоритмов, которые используются для генерации текста.

Задача состоит в том, чтобы четко сформулировать правила музыки или, по крайней мере, музыкального языка [7]. Описание процесса композиции с помощью явных и формальных правил может быть использовано для программирования, автоматизации и моделирования определенных аспектов процесса музыкальной композиции.

1.4.3 Методы искусственного интеллекта

Применение методов искусственного интеллекта для генерации музыки имеет большое значение в генерации произведений. Музыка организована на множестве взаимосвязанных уровней абстракции, что делает ее своеобразным испытательным полигоном для методов искусственного интеллекта [8].

Первым этапом в решении множества задач с помощью средств искусственного интеллекта (ИИ) является сведение задачи к базовой проблеме, которая решается средствами ИИ. Одной из таких проблем является предсказание последовательности, которое используется в приложениях для перевода и обработки естественного языка. Нашу задачу по генерации музыки можно свести к проблеме предсказания последовательности, при этом предсказание будет выполняться для последовательности музыкальных нот.

Системы обучаются на основе большого количества музыкальных произведений или по некоторым правилам гармонии. Например, система HARMONET обучалась на основе правил гармонии И. Баха.

1.4.4 Биологические методы

Суть биологических методов в том, что они напоминают биологические процессы. В них входят, например, генетический алгоритм и клеточные автоматы.

1.5 Вывод

В данном разделе была обоснована актуальность задачи, рассмотрены объекты, с которыми работают алгоритмы, а также описаны основные проблемы генерации музыки и группы, на которые делятся методы генерации.

2 Классификация существующих методов

В данном разделе будут рассмотрены методы генерации музыки, а также проведено их сравнение и оценка.

2.1 Методы генерации музыкальных произведений полифонического склада

Далее будут приведены и описаны наиболее популярные методы генерации музыки.

2.1.1 Марковские цепи

Цепь Маркова - это модель, описывающая последовательность возможных событий. Эта последовательность должна удовлетворять предположению Маркова - вероятность следующего состояния зависит от предыдущего состояния, а не от всех предыдущих состояний в последовательности.

Это может звучать как упрощение реальных дел. Например, для применения цепочки Маркова для прогнозирования погоды нам нужно сделать предположение, что погода завтра зависит только от текущей погоды, и делать вид, что других факторов, таких как время года, например, нет.

Несмотря на это упрощение во многих случаях, мы сможем генерировать полезные прогнозы, но в то же время мы сможем быстрее решать нашу задачу, делая ее менее затратной в вычислительном отношении.

Модели цепей Маркова имеют множество применений в области финансов, обработки естественного языка и в любом месте, где у вас есть данные временных рядов.

К музыкальной композиции можно применить цепи Маркова [6], поскольку музыкальное произведение можно рассматривать как последовательность состояний, каждое из которых представляет из себя ноту определенной длины. Поскольку доступные ноты не бесконечны, варианты длины также не

бесконечны, то категории состояния также должны быть конечны. Таким образом, цепи Маркова могут быть построены из предыдущих музыкальных произведений разных жанров и служить основой для алгоритма обучения для принятия вероятностных решений и создания новых музыкальных произведений в том же жанре. Использование Марковских цепей n -го порядка позволяет сгруппировать отдельные ноты вместе, создавая различные шаблоны и последовательности.

2.1.2 Контекстно-зависимая грамматика

Контекстно-зависимая грамматика (КЗ-грамматика, контекстная грамматика) [9] – частный случай формальной грамматики, у которой левые и правые части всех продукций могут быть окружены терминальными и не терминальными символами.

Суть использования кз-грамматик можно описать следующим примером. Изначально имеется строка: *ABCDEFGHIKFLHLEFJ*. Для данной строки необходимо построить грамматику.

В качестве примера, возьмем символ F. Для данной буквы мы формируем правило, которое будет указывать на то, какая буква должна идти после F. Согласно начальной строке, после F могут идти буквы G, H и J. Поскольку нельзя однозначно выбрать одну из букв, нужно расширить контекст буквами, которые окружают F. Взяв по одной букве перед F, имеем следующие наборы букв – EF и KF.

Теперь после KF идет только буква H, значит было получено первое правило: $KF \rightarrow H$. Но после EF могут идти G и J, тогда снова расширяем контекст и получаем: DEF и LEF. Тогда конечный набор правил для буквы F имеет вид: $KF \rightarrow H$, $DEF \rightarrow G$, $LEF \rightarrow J$.

Получаем набор правил, значит можно генерировать строку по заданной.

При этом возникает проблема: если генерировать новую последовательность по строго заданным правилам, то будет получаться исходная строка. Поэтому нужно брать промежуточные правила или конечные. Выбор осуществляется некоторым случайным образом, тогда строка в каждом из случаев будет уникальной.

Данный метод генерации применим не только для строк, но и для ме-

лодий, поскольку букву можно заменить на определенную ноту или даже аккорд, при этом принцип генерации останется тот же.

КЗ-грамматика хорошо подходит для задачи генерации музыкальной партитуры в виде нот, однако требует определения чётких формальных правил построения композиции, что является крайне трудоёмким процессом.

2.1.3 Л-системы

Л-система или система Линденмайера – это параллельная система перезаписи и тип формальной грамматики.

Правила грамматики Л-системы применяются итеративно, начиная с начального состояния. Максимально возможное количество правил применяется одновременно за итерацию. Тот факт, что на каждой итерации используется как можно больше правил, отличает Л-систему от формального языка, порожденного формальной грамматикой, которая применяет только одно правило на итерацию. Если бы продукционные правила применялись только по одному за раз, можно было бы довольно просто создать язык, а не Л-систему. Таким образом, Л-системы являются строгими подмножествами языков.

Работа Л-системы [10] начинается с определения трех наборов параметров – алфавита (список символов, которыми будет оперировать система, как на входе, так и на выходе, в нашем случае, например, ноты), набора правил (например, классических законов гармонии) и аксиомы (символ или строка символов для начального ввода). Л-системы относятся к экспертным системам, основанным на знаниях и заданных наборах правил. К таким системам также можно отнести уже упоминавшиеся Марковские цепи, в которых вероятность перехода от одной ноты к другой вычисляется на основании матрицы вероятностных переходов, исходя из заранее определенных статистических правил.

В основе работы Л-систем лежит набор правил замещения, рекурсивно применяющийся на начальную строку символов и интерпретирующий конечную строку, как структурные элементы организма. Правила замещения определяют, как каждый конкретный символ в текущем поколении должен быть перемещен. Применение Л-систем для генерации алгоритмических компози-

ций подразумевает использование вместо символов определенных музыкальных параметров.

2.1.4 Клеточные автоматы

Клеточные автоматы – это дискретные динамические системы, поведение которых полностью определяется в терминах локальных зависимостей. Пространство представлено равномерной сеткой, каждая ячейка или клетка которой содержит несколько битов данных; время идет вперед дискретными шагами, а законы мира выражаются единственным набором правил, скажем, небольшой справочной таблицей, по которой любая клетка на каждом шаге вычисляет своё новое состояние, основываясь на состояниях её близких соседей. Таким образом, законы системы являются локальными и повсюду одинаковыми.

Самым известным воплощением клеточных автоматов для генерации алгоритмических композиций является CAMUS. CAMUS генерирует мелодию на основе двух клеточных автоматов: игра в «жизнь» [11] Джона Конвеема и Demon Cyclic Space Больцмана. CAMUS оперирует трехмерными версиями обоих автоматов параллельно. При каждом шаге живые ячейки Игры в жизнь используются для определения четырехнотных аккордов, в то время, когда координаты живых клеток определяют интервалы сыгранных нот. Базовый тон выбирается стохастическими методами (Марковскими цепями), и координата x живой клетки определяет интервал до следующей самой высокой ноты, координата y — дает полутоновый интервал, и, наконец, координата z дает последний интервал от второй самой высокой ноты до самой высокой.

2.1.5 Генетический алгоритм

Генетический алгоритм (ГА) [12] — это эвристический алгоритм поиска, используемый для решения задач оптимизации и моделирования путём случайного подбора, комбинирования и вариации искоемых параметров с использованием механизмов, напоминающих биологическую эволюцию. В общем смысле работа генетического алгоритма начинается с применения экви-

валента биологического образования новых генов на пространство случайно распределенных решений для нахождения в итоге оптимального набора.

Решения представлены хромосомами, а строки аллель — строками чисел, и задача рекомбинации генов заключается в создании новых аллелей из аллели, взятых от родительских хромосом посредством применения генетических операторов, в большинстве случаев это — мутация и скрещивание. Перебирание хромосом продолжается до достижения определенного условия экстремума.

Генетические алгоритмы в задаче алгоритмической композиции разделяются по виду использованной *фитнесс-функции* — степень приспособленности хромосом может быть оценена исходя из заранее заданных определенных условий, либо может быть непосредственно человеком при прослушивании и субъективной оценке.

Первыми успешными исследованиями в области применения генетических алгоритмов в генеративной музыке можно считать изыскания Джона Бильса — профессора Рочестерского института технологий (Нью-Йорк, США). Он использовал ГА для имитации джазовой импровизации в собственном программном обеспечении GenJam [12] (Genetic Jammer — Генетический Джэммер). Данная программа читает заранее подготовленные с помощью программы файлы, включающие партию аккордов пианино, баса, ритм-секции, и генерирует соло. Оценка происходит посредством человеческого восприятия. Посредством команд «g» или «b» («good» or «bad») слушатель оценивает сгенерированные куски как удачные или нет.

2.1.6 Генеративно-состязательные сети

Генеративно-состязательная сеть (GAN) [13] — нейросеть, которая объединяет в себе две нейросети: генератор и дискриминатор, которые действуют друг против друга.

Задача генератора — создать образ, а дискриминатор — отличает настоящее от поддельного, то есть решает: относится ли образ к набору тренировочных или нет. Если дискриминатор распознал объект как подделку, то генератор снова пытается «обмануть» дискриминатора. Таким образом, нейросеть обучается благодаря «противостоянию» двух нейросетей, входящих

в нее. Использование этой структуры позволяет обучать глубокие генеративные модели без дорогостоящих нормализующих констант. При этом этот метод доказал, что он позволяет создавать очень реалистичные выборки данных. Также для обучения в нейросеть загружается множество примеров музыкальных произведений, как современных, так и произведений классической музыки.

2.1.7 Рекуррентные нейронные сети

Рекуррентные нейронные сети (RNN) [14] – вид нейронных сетей, где связи между элементами образуют направленную последовательность. Благодаря этому появляется возможность обрабатывать серии событий во времени.

Рекуррентные нейронные сети часто используются для моделирования последовательностей данных. Сети могут использовать свою внутреннюю память для обработки. Эти модели обычно обучаются с использованием критерия максимального правдоподобия. Например, для языкового моделирования их обучают предсказывать следующую лексему в любой точке последовательности, то есть моделировать условную вероятность следующей лексемы при заданной последовательности предшествующих лексем. Путем выборки из этого условного распределения можно генерировать достаточно реалистичные последовательности. Выборка нетривиальна, и вы обычно прибегаете к поиску для создания хороших последовательностей в таких задачах, как машинный перевод. RNN использовались для моделирования музыки, используя при этом символьное представление.

Вариант использования – **LSTM сети** [5], которые способны распознавать и кодировать долгосрочные шаблоны. Они крайне полезны в случае, когда сеть должна запоминать информацию в течение длительного периода времени, что и применяется при генерации музыки.

2.2 Сравнение и оценка методов

На основании описанных выше методов для генерации музыки критерием оценки качества был выбран статистический опрос по музыкальному тесту Тьюринга.

Тест Тьюринга – эмпирический тест, который заключался в следующем: человек взаимодействует с одним компьютером и одним человеком. На основании ответов на вопросы он должен определить, с кем он разговаривает – с человеком или компьютерной программой. Задача компьютерной программы – ввести человека в заблуждение, заставив сделать неверный выбор. Все участники теста не видят друг друга. Если судья не может сказать определённо, кто из собеседников является человеком, то считается, что машина прошла тест. При музыкальном тесте Тьюринга человеку предлагается отличить музыку, написанную человеком, и музыку, которую написала машина.

В нашем опросе было предложено по 2 музыкальные композиции для каждого из 7 методов – одна композиция написана человеком, другая – получена с использованием метода генерации музыки. Если человек делал неправильный выбор или сомневался – считалось, что сгенерированный фрагмент прошёл проверку. Вычислялось процентное соотношение пройденных проверок к их общему количеству для каждого отдельно взятого метода генерации. Результаты опроса приведены в таблице 2.1.

Таблица 2.1 – Результаты музыкального теста Тьюринга для каждого из методов

Метод	Процент успеха
Цепочка Маркова	71%
КЗ-грамматики	91%
Л-системы	80%
Клеточные автоматы	74%
Генетический алгоритм	67%
Генеративно-состязательные сети	87%
Рекуррентные нейронные сети	83%

Как видно из результатов, самым точным методом оказался метод с использованием контекстно-зависимых грамматик. Дело в том, что метод способен генерировать музыку высокого качества, если будет правильно подобран набор правил для генерации, что является крайне трудной задачей. Также высокий показатель у методов, основанных на нейронных сетях. За счет долгого обучения они способны показать высокое качество генерации музыкального фрагмента.

2.3 Вывод

В данном разделе были представлены и описаны методы генерации музыки, а также проведено их сравнение и оценка их качества на основании статистического критерия.

Заключение

В ходе данной работы были классифицированы некоторые алгоритмы генерации музыкального фрагмента с использованием современных технологий. В качестве самого точного алгоритма для дальнейшего исследования был выбран алгоритм, позволяющий генерировать музыку полифонического склада на основе контекстно-зависимых грамматик, поскольку он на основе статистического опроса оказался самым точным.

Цель, которая была поставлена в начале работы была достигнута, а также в ходе выполнения научно-исследовательской работы были решены следующие задачи:

- были описаны виды проблем, возникающих при генерации музыки;
- были описаны методы, которые позволяют получить музыкальный фрагмент;
- проведен сравнительный анализ по выделенному критерию для всех методов генерации;
- выбран наиболее точный метод генерации музыкального произведения.

Список литературы

- [1] Vandana T. Music Mood Categorization: A Survey [Электронный ресурс]. URL: https://www.researchgate.net/publication/334096365_Music_Mood_Categorization_A_Survey. 2020. (Дата обращения: 21.01.2022).
- [2] Freisleben B. The Neural Composer: A Network for Musical Applications [Электронный ресурс]. 1992. URL: <https://www.sciencedirect.com/science/article/pii/B9780444894885501809>. (Дата обращения: 21.01.2022).
- [3] Briot J.-P. Deep Learning Techniques for Music Generation – A Survey [Электронный ресурс]. 2020. URL: <https://hal.sorbonne-universite.fr/hal-01660772/document>. (Дата обращения: 21.01.2022).
- [4] Briot J.-P. Music Generation by Deep Learning– Challenges and Directions [Электронный ресурс]. 2018. URL: <https://arxiv.org/pdf/1712.04371.pdf>. (Дата обращения: 21.01.2022).
- [5] Mangal S. LSTM Based Music Generation System [Электронный ресурс]. 2019. URL: <https://arxiv.org/pdf/1908.01080.pdf>. (Дата обращения: 21.01.2022).
- [6] Shapiro I. Markov Chains for Computer Music Generation [Электронный ресурс]. URL: <https://scholarship.claremont.edu/cgi/viewcontent.cgi?article=1848&context=jhm>. 2020. (Дата обращения: 23.01.2022).
- [7] Mehta R. Music Generation using Parametric Grammar [Электронный ресурс]. 2018. URL: <https://www.ijert.org/research/music-generation-using-parametric-grammar-IJERTCONV3IS01025.pdf>. (Дата обращения: 21.01.2022).
- [8] Yang Xin-She. Nature-Inspired Computation and Swarm Intelligence. Academic Press, 2020. С. 217–245.
- [9] Eibensteiner L. Procedural Music Generation with Grammars [Электронный ресурс]. 2018. URL: https://cescg.org/cescg_submission/procedural-music-generation-with-grammars/. (Дата обращения: 21.01.2022).

- [10] Worth P. Growing Music: musical interpretations of L-Systems [Электронный ресурс]. 2018. URL: <https://www-users.cs.york.ac.uk/susan/bib/ss/nonstd/eurogp05.pdf>. (Дата обращения: 25.01.2022).
- [11] Adamatzky A. Game of Life Cellular Automata [Электронный ресурс]. URL: https://www.researchgate.net/publication/253231764_Game_of_Life_Cellular_Automata. 2010. (Дата обращения: 27.01.2022).
- [12] Farzaneh M. GGA-MG: Generative Genetic Algorithm for Music Generation [Электронный ресурс]. 2020. URL: <https://arxiv.org/pdf/2004.04687.pdf>. (Дата обращения: 19.01.2022).
- [13] Li S. Automatic Melody Composition Using Enhanced GAN [Электронный ресурс]. URL: https://www.researchgate.net/publication/336001437_Automatic_Melody_Composition_Using_Enhanced_GAN. 2019. (Дата обращения: 21.01.2022).
- [14] Gunawan A. A. S. Automatic Music Generator Using Recurrent Neural Network [Электронный ресурс]. URL: https://www.researchgate.net/publication/342444314_Automatic_Music_Generator_Using_Recurrent_Neural_Network. 2020. (Дата обращения: 21.01.2022).