СТАТЬИ

УДК 004.81

# МЕТОДЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА В ГЕНЕРАЦИИ АЛГОРИТМИЧЕСКИХ МУЗЫКАЛЬНЫХ КОМПОЗИЦИЙ

**Бурякова О.С., Решетникова И.В., Черкесова Л.В.**

*ФГБОУ ВО «Донской государственный технический университет», Ростов-на-Дону, e-mail: chia2002@inbox.ru*

Предлагаемая статья посвящена процессу генерации музыкальных композиций искусственными нейронными сетями по заранее заданным параметрам. Обучение нейронных сетей проведено по образцам классической (джазовой, рок и др.) музыки и по изображениям. В работе проанализированы существующие сегодня в мире системы нейросетевой генерации музыки, а также исследованы алгоритмы и программные средства, разработанные в этой области, как зарубежные, так и отечественные. Разработана нейросетевая encoder-decoder архитектура, для построения которой авторами были применены методы и механизмы генеративных моделей обработки мультимодальной информации и нейронные сети. Исследован механизм внимания, разработан кодировщик изображений, предложен набор данных (датасет) и применён алгоритм эволюционного обучения нейронной сети. Проведен анализ структуры, алгоритмическое и программное конструирование, реализованы алгоритмы его модулей, исследованы современные библиотеки и функции программного обеспечения, связанные с разработкой и оценкой качества сгенерированного музыкального ряда. Реализовано программное средство, предназначенное для алгоритмической генерации музыкальных композиций на основе изображений по заданным параметрам обучения нейронной сети. Для его создания использована программная платформа в виде фреймворка машинного обучения Pytorch языка Python. Полученные в виде звуковых музыкальных файлов композиции, сгенерированные искусственной нейронной сетью, могут быть полезны и оказать существенную помощь в работе композиторов, звукорежиссёров или музыкальных оформителей в самых различных ситуациях: при написании саундтреков кино-, теле- или мультфильмов, музыки для спектаклей и театральных постановок, при озвучивании экспозиций в мультимедийных музеях, при звуковом оформлении мероприятий разного формата, выставок, концертов, шоу-программ, онлайн- и офлайн-мероприятий, проводимых в маркетинговых целях, и др. Осуществлена демонстрация работы программного средства в виде прослушивания звуковых midi-файлов, содержащих музыкальные композиции, сгенерированные нейронной сетью по заданным параметрам.

**Ключевые слова: искусственные нейронные сети, нейросетевая генерация музыки, генеративное моделирование, мультимодальная информация, алгоритмы эволюционного обучения, кодировщик изображений, датасет**

# METHODS OF ARTIFICIAL INTELLIGENCE IN THE GENERATION OF ALGORITHMIC MUSICAL COMPOSITIONS

**Buryakova O.S., Reshetnikova I.V., Cherkesova L.V.**

*Don State Technical University, Rostov-on-Don, e-mail: chia2002@inbox.ru*

The proposed article is devoted to the process of musical compositions generating by artificial neural network according to predefined parameters. Neural network training is conducted based on samples of classical (jazz, rock, etc.) music, grounded on images. The paper analyzes the existing systems of neural network music generation in the world, investigates algorithms and software developed in this field, both foreign and domestic. Neural network encoder–decoder architecture has been developed, for the construction of which the authors used methods and mechanisms of generative models of multimodal information processing and neural networks. The mechanism of attention is investigated, image encoder is developed, dataset is proposed and the algorithm of evolutionary training of neural network is applied. The analysis of the structure, algorithmic and software design is carried out, algorithms of its modules are implemented, modern libraries and software functions related to the development and evaluation of the quality of the generated musical compositions are investigated. Software tool designed for algorithmic generation of musical compositions based on images has been implemented. To create the software tool, software platform Pytorch – the machine-learning framework of Python programming language was used. The result of research work is development of software tool designed for neural network generation of musical compositions according to the specified parameters of neural network training. Compositions obtained in the form of sound music files generated by an artificial neural network can be useful and provide significant assistance in the work of composers, sound engineers or music designers in a variety of situations. These cases are when writing soundtracks of movies, TV or cartoons, music for performances and theatrical productions, when voicing expositions in multimedia museums, when sound design events of various formats, exhibitions, concerts, show programs, online and offline events held for marketing purposes, etc. Demonstration of the software operation was carried out, in the form of listening to audio midi-files containing musical compositions generated by artificial neural network according to specified parameters.

**Keywords: artificial neural networks, neural network music generation, generative modeling, multimodal information, evolutionary learning algorithms, image encoder, and dataset**

Музыка представляет собой неотъемлемую часть человеческой культуры, существующую с самых ранних периодов человеческой цивилизации. Вопрос о том, поддаётся ли этот процесс алгоритмизации и смогут ли компьютерные технологии запечатлеть творческий процесс, занимал учёных на протяжении многих десятилетий. Любую музыку можно представить в виде последовательности закономерных данных,

без потерь в содержании. Это стало возможным ещё с древних времен, с появления записи музыки в виде нот, из которых состоит и записывается любое музыкальное произведение. Нотные партитуры могут быть представлены в памяти компьютера в виде последовательности бинарных векторов.

Благодаря нотам и нотным сочетаниям, музыка представляет собой неисчерпаемый источник для исследований с помощью искусственного интеллекта. Глубинное машинное обучение на сегодняшний день является актуальным инструментом для построения сложных моделей нейронных сетей, при исследовании и моделировании творческой деятельности.

За последние годы в области генеративных моделей достигнут огромный прогресс [1]. Одна из целей генеративного моделирования состоит в охвате основных аспектов данных и генерировании новых экземпляров, неотличимых от истинных данных. Гипотеза состоит в том, что, если научиться производить данные, то появится возможность узнать их главные особенности. Благодаря успехам в развитии искусственных нейронных сетей, с помощью компьютера можно создавать сложные музыкальные композиции для большого оркестра [1, 2]. Недостатком этого процесса является малое число существующих архитектур и ограниченный контекст, которым может оперировать нейронная сеть, способная уловить характерные зависимости на протяжении десятков тысяч нот; что необходимо для создания сложной, согласованной и продолжительной музыкальной композиции.

При создании музыки естественные нейронные сети используют мультимодальную информацию, принцип которой основан на соединении нескольких *модусов восприятия* информации в процессе коммуникации. Отсутствие мультимодальности у существующих генеративных моделей также препятствует созданию сложных музыкальных произведений.

Дело в том, что естественные нейроны способны хранить доменно-независимую информацию об объекте – в виде текста, изображения, звука, запаха, вкуса, веса и осязания.

При этом нейронная сеть способна генерировать музыку на основе изображений, для чего потребуется усваивать концепции вне зависимости от модальности, что хорошо согласуется с последними исследованиями в области искусственного интеллекта [3].

Создание длинных музыкальных произведений – сложная задача, поскольку музыка содержит структуру в различных временных масштабах, от миллисекундных таймингов до мотивов, фраз и повторения целых разделов. Используемая авторами нейронная сеть *Music Transformer* способна генерировать музыку с улучшенной долгосрочной когерентностью [4].

В разработке также применена глубокая нейронная сеть *MuseNet*, способная генерировать 4-минутные музыкальные композиции с 10 различными инструментами и сочетать различные музыкальные стили: от Моцарта до джаза, от Чайковского до рэпа и рок-н-ролла. Эта нейронная сеть не была запрограммирована с нашим пониманием музыки, но вместо этого обнаружила закономерности гармонии, ритма и стиля, научившись предсказывать следующий элемент (*токен*) в сотнях тысяч midi-файлов. *MuseNet* использует крупномасштабную модель *трансформера,* универсальную технологию, обученную предсказывать следующую музыкальную фразу в звуковой аудиопоследовательности, что позволяет применять пересчитанные и оптимизированные ядра датасета *Sparse Transformer* [4, 5] для обучения 72-слойной сети с 24 головками *механизма внимания* в контексте 4096 токенов.

Алгоритм механизма внимания является новым качественным улучшением в нейронной сети *Transformer*. Он позволяет извлекать модель регулярности во всём множестве слоев музыкального произведения и на их основе строить прогноз. Главный принцип работы сети основан на соответствии каждому выходному элементу каждого входного. Динамически просчитывая вес между элементами в соответствии с внешними условиями, сеть строит *матрицу внимания*. *Головки внимания* – это своеобразные точки фокусировки, которыми нейронная сеть исследует музыкальные регулярности. Этот длинный контекст является одной из причин способности обученной нейронной сети запоминать долгосрочную структуру композиции [5].

Для обработки мультимодальной информации использована нейронная сеть архитектуры *Perceiver*, которая может получать и классифицировать входящие данные нескольких типов – облако точек, аудио и изображения. Для этой цели модель глубокого обучения нейронной сети основана на преобразователях – *механизмах внимания*. Недостатком использования таких преобразователей является квадратичное количество операций, необходимых для алгоритмов. Так, обработка изображения размером 224×224 пикселов может привести к $224^2$ операциям. Это более 50 000 операций – огромные вычислительные затраты. Для сокращения числа операций уровень «*механизма внимания*» заменён слоем «*механизма перекрестного внимания*» в преоб-

разователе [6], что привело к уменьшению сложности алгоритма до уровня линейной сложности. Входные данные, используемые для вычисления *перекрестного внимания*, преобразованы в массив байтов, что означает независимость модели от типа данных. Прорыв в этой модели нейронной сети заключается в том, что она работает с любыми типами входных данных, тогда как *сверточные нейронные* сети *работают* только с изображениями [7].

Целью исследования является разработка программного средства с комплексом модулей нейронных сетей, способного генерировать музыкальные композиции по заданным параметрам на основе произвольных изображений. *Объектом исследования* выступает процесс генерации музыки искусственной нейронной сетью на основе изображений. *Предметом исследований* являются механизмы иерархического внимания, внутренней памяти, а также обработка мультимодальной информации. Для достижения поставленной цели авторами изучены существующие в мире современные нейросетевые решения, исследованы основные механизмы и методы генерации музыки; разработаны механизмы иерархического внимания; создан кодировщик изображений для генерации музыки; выбран набор данных (датасет) для обучения нейронной сети, на котором проведено её обучение, в том числе с целью оценки соответствия изображения и музыки; проведено алгоритмическое конструирование механизма эволюционного обучения нейронной сети для генерации музыки; создано программное средство для алгоритмической генерации музыкальных композиций на основе изображений, на программной платформе – фреймворке машинного обучения Pytorch, на языке программирования Python 3.

**Материалы и методы исследования**

Материалы и методы исследования связаны с нейросетевой генерацией музыки. Авторами применён новый метод искусственного интеллекта с названием *«генетическое программирование»,* генерирующий собственные музыкальные материалы и формирующий свою собственную грамматику. В этой модели нейронной сети человек (композитор) должен запрограммировать функцию «*критика*». Это обученная нейронная сеть, предназначенная для прослушивания многочисленных автоматически созданных музыкальных композиций – выходных данных на различных этапах обработки, чтобы решить, какие из них подходят (или не подходят) для окончательного результата. Вопрос о том, какие результаты генерирования музыкальных композиций следует отбросить, а какие сохранить, решает человек – композитор.

*Инструментом создания* программного средства генерирования музыкальных композиций является программная платформа *PyTorch* – фреймворк машинного обучения для языка Python с открытым исходным кодом, созданный на базе *Torch*. Он используется для решения задач искусственного интеллекта и оптимален для данной задачи исследования. Вокруг этого фреймворка выстроена экосистема из библиотек, созданных сторонними разработчиками *PyTorch – LightningPyTorch* и *Lightning*, упрощающих процесс обучения моделей нейронных сетей.

Так, библиотека *Pyro* предназначена для вероятностного программирования, *Flair* – для обработки естественного языка, и *Catalyst* – для обучения моделей многослойных нейронных сетей DL (Deep Learning) и RNN (Recurrent Neural Network) [8].

Важным инструментом обучения нейронных сетей является «*механизм внимания»* – вектор, представляющий собой выходы слоя с использованием функции *Softmax*. Это интерфейс, сформированный заданными параметрами, который можно подключить там, где программист сочтёт нужным, качественно улучшает анализ регулярности в музыкальных произведениях [6–8].

Генеративный предварительно обученный трансформер 3 (GPT3) генерирует звук с использованием предварительно обученных алгоритмов. Перед этим им уже были переданы все данные, необходимые для выполнения их задач. В частности, были переданы около 570 ГБ звуковой информации, собранной путём обхода Интернета – общедоступный набор данных, известный как CommonCrawl, а также другие файлы, выбранные OpenAI. Количество весов, которые алгоритм динамически хранит в своей памяти и использует для обработки каждого запроса, составляет 175 миллиардов. Качество музыки, генерируемой GPT–3, настолько высоко, что его трудно отличить от композиции, созданной человеком, который имеет как преимущества, так и недостатки. Обучение происходило на датасете MAESTRO и midi-файлах из интернета. Результаты представлены отдельным файлом со звуковым сопровождением [9].

В работе [10] представлена новая архитектура под названием *Transformer*, использующая *механизм внимания*, предназначенная для преобразования одной последовательности данных любого типа в другую последовательность, с помощью кодировщика и декодировщика (рис. 1).

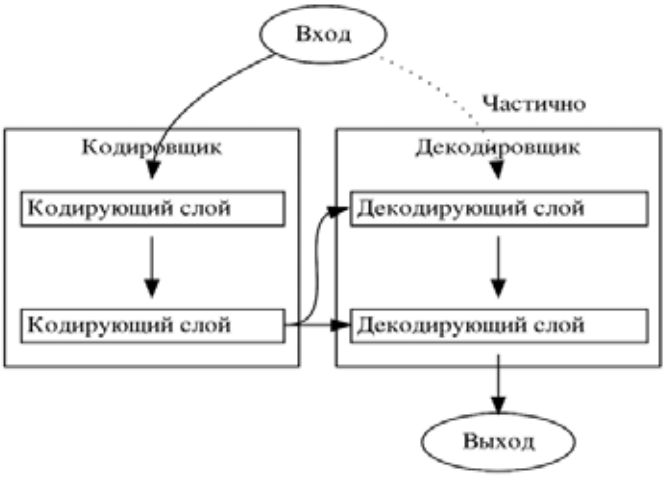
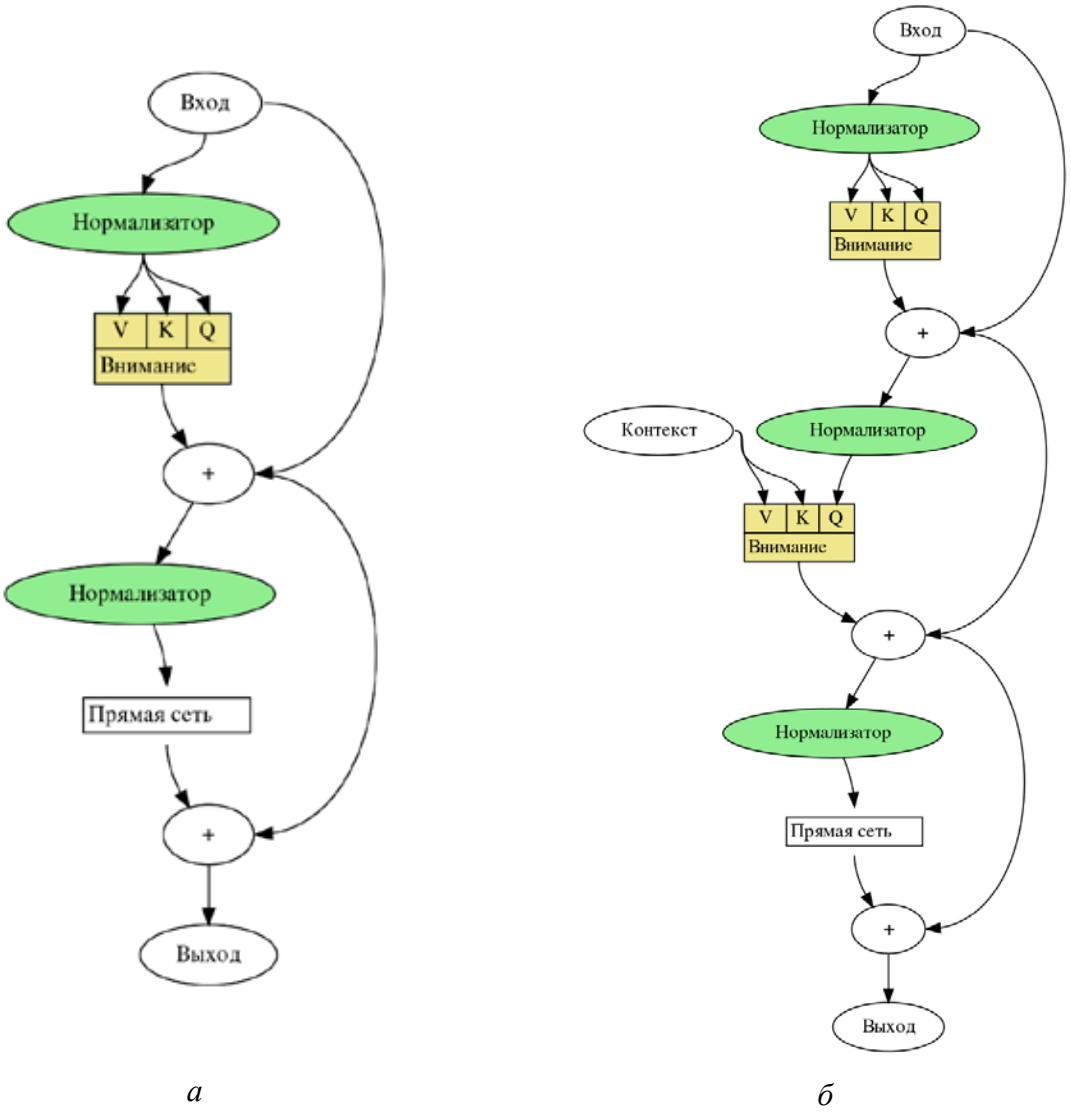



*Рис. 1. Архитектура трансформера*




*а* *б*

*Рис. 2. Архитектура кодировщика (а) и декодировщика (б)*

И кодировщик, и декодировщик состоят из модулей, которые могут пересекаться друг с другом несколько раз (рис. 1). Модули состоят из *слоев внимания* и *сетей прямой связи*. Входы и выходы встраиваются в n-мерное пространство. Каждый кодировщик состоит из двух основных компонентов: *механизма самосознания* и *нейронной сети обратной связи*.

Механизм самосознания принимает входные кодировки от предыдущего кодировщика и взвешивает их релевантность друг другу, чтобы сгенерировать выходные кодировки. Нейронная сеть обратной связи обрабатывает каждое выходное кодирование индивидуально. Выходные кодировки затем передаются следующему кодировщику в качестве входных данных, а также декодировщикам. Первый кодировщик воспринимает вложения входной последовательности и позиционную информацию в качестве входных данных, а не кодировок. Информация об их положении необходима для того, чтобы трансформер использовал порядок последовательности (рис. 2, а). Каждый декодировщик состоит из трёх компонентов: *механизма самосознания, механизма внимания* над кодировками и *нейронной сети обратной связи*. Декодировщик работает аналогично кодировщику, но имеет дополнительный встроенный *механизм внимания*, извлекающий нужную информацию из кодировок, генерируемых кодировщиками (рис. 2, б).

Так же как и первый *кодировщик*, первый *декодировщик* принимает позиционную информацию и вложения выходной последовательности в качестве входных данных.

Трансформер не должен использовать текущий или будущий выход для прогнозирования выхода, поэтому выходная последовательность должна быть частично замаскирована, чтобы предотвратить обратный поток информации. За последним декодировщиком следует конечное линейное преобразование, а также слой Softmax, чтобы получить выходные вероятности [10].

Для генерации алгоритмов музыки авторами использован *параллельный алгоритм выбора бинарного турнира*. Эволюционные эксперименты начинаются с популяции случайно инициализированных генотипов и проводят 100 000 бинарных турниров.

Каждый генотип кодирует всю информацию, необходимую для генерации музыки, а именно входную строку *рекурентной нейронной сети* и её параметры. Бинарный турнир состоит из оценки двух генотипов, с последующей заменой генотипа с более низкой приспособленностью на *мутантную копию* генотипа с более высокой приспособленностью.

Оценка генотипа включает в себя сначала построение музыки из генотипа, а затем оценку музыки с помощью *двойного кодировщика*, обусловленного входным изображением. Общий процесс включает в себя, для начала, получение *случайной популяции генотипов* (рис. 3).

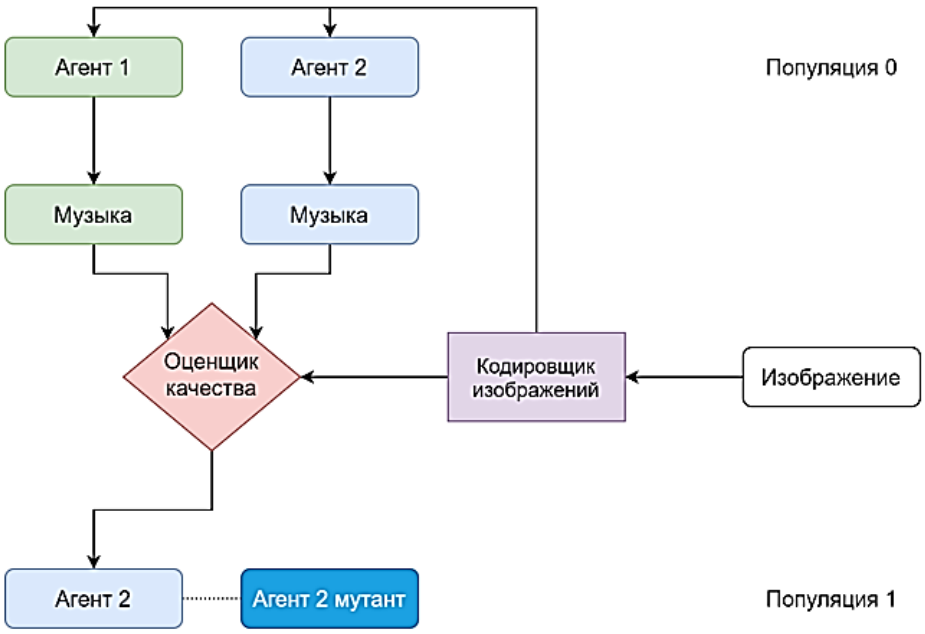



*Рис. 3. Схема эволюционного алгоритма*

Оцениваются два случайных генотипа. Это делается путём перевода генотипов во входную строку и в значения весов *рекуррентной нейросети*, использовавшихся этой системой для получения изображения. Затем сгенерированная музыка прослушивается *оценщиком качества*, который также получает на вход изображение и выдаёт оценку пригодности музыкальной композиции. Эта функция используется для определения победителя (какой из двух генотипов выиграл соревнование), а затем победивший генотип мутирует, и мутантная копия возвращается в популяцию, переписывая проигравший генотип, который должен быть удалён [7, 8].

Для генерации музыки используются *генеративная* модель и *иерархический трансформер* скрытых состояний входных токенов. Трансформер, основанный на *механизме внимания*, но не являющийся рекуррентным, позволяет обучить нейронную сеть более эффективно, так как рекуррентные нейронные сети, как и модели на основе трансформеров, имеют ограничения на длину входной последовательности. При слишком длинной последовательности качество обучения ухудшается. Для решения этой проблемы был применён метод *иерархического внимания*, специально созданный для задач классификации длинных последовательностей.

*Архитектура трансформера* (рис. 4), основанная на *механизме внимания*, применяется во многих задачах глубокого обучения моделей нейронных сетей. В различных вариантах архитектуры трансформера [10] используется предварительная подготовка кодировщиков, декодировщиков и архитектуры «*encoder-decoder*». При генерации последовательностей, включая звуковые [10], производительность работы нейронной сети значительно повышается за счёт инициализации параметров предварительно обученных моделей (рис. 5).

*Иерархическое обучение* модели трансформеров [10] в задачах генерации длинных последовательностей применено для повышения производительности, для чего была модифицирована стандартная архитектура, с добавлением функции «*иерархического внимания*». Использовано 12 слоёв *кодировщика* и *декодировщика* для полностью подключенных сетей *прямой связи* и 16 головок *механизма внимания*, как в кодировщике, так и в декодировщике. Добавлен дополнительный слой кодировщика, который обрабатывает вложения токенов, вставленные при предварительной обработке данных. Это слой *иерархического кодировщика*, производящий другой уровень контекстных представлений для каждого из токенов. Применение одного иерархического уровня работает лучше, хотя может использоваться и несколько иерархических уровней. Каждый слой декодировщика выполняет «*функцию самовнимания*» над заранее созданными токенами, а затем следит за выходами финального слоя кодировщика.

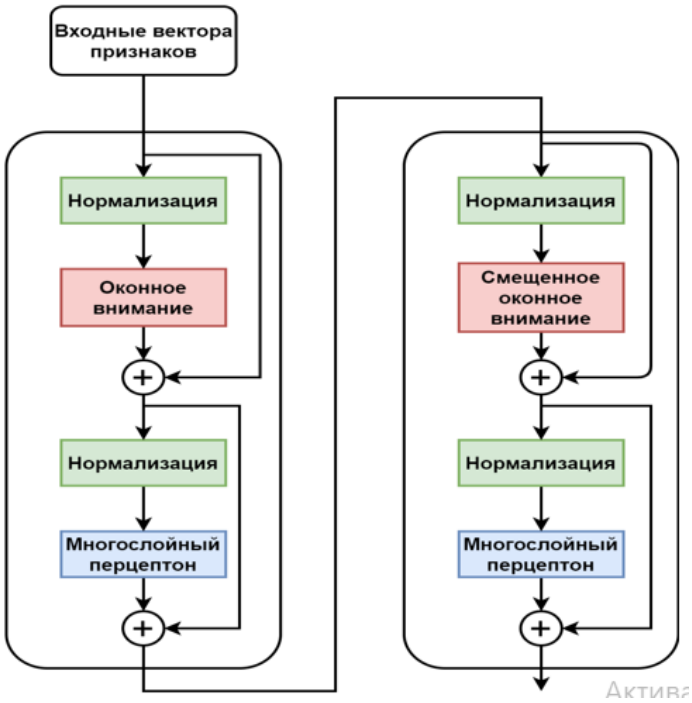



*Рис. 4. Блок трансформера*

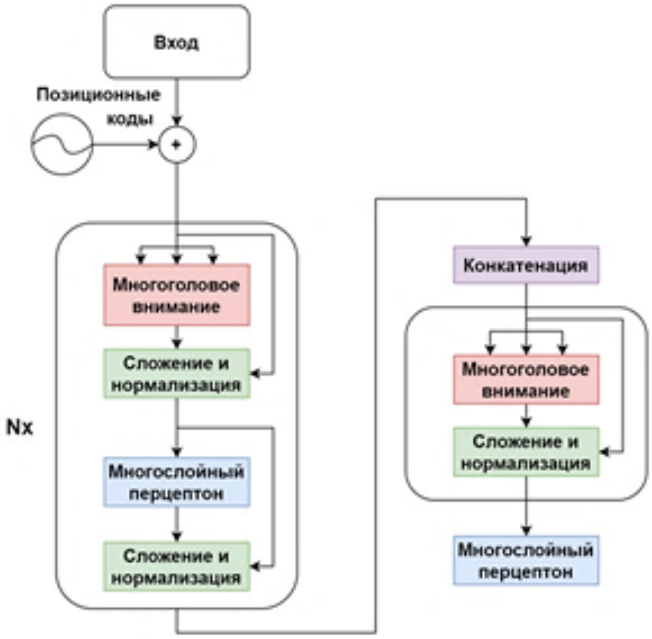



*Рис. 5. Генератор музыки*

Добавлен *модуль внимания*, внедряющий токены из слоя иерархического кодировщика. Обучение нейронной сети проходило на датасете MAESTRO и midi-файлах из сети Интернет.

*Для кодирования изображений* использована нейронная сеть *Swin-Transformer*, которая строит иерархическое представление, начиная с небольших участков, и постепенно объединяет соседние участки в более глубокие слои трансформера. С помощью иерархических карт объектов модель Swin Transformer [9] использует передовой метод *плотного прогнозирования – пирамидальные сети объектов* (FPN или U-Net). Линейная вычислительная сложность достигается за счёт локального вычисления *функции самосознания* в неперекрывающихся окнах, разделяющих изображение. Количество патчей в каждом окне фиксировано, и, таким образом, сложность становится линейной по размеру изображения. Это делает сеть Swin Transformer основой для задач компьютерного зрения, в отличие от архитектур на основе трансформеров, которые создают карты объектов с одним разрешением и имеют квадратичную сложность (рис. 6).

Сначала входное RGB-изображение разбивается на неперекрывающиеся патчи, с помощью модуля разделения патчей, такого как ViT. Каждый патч обрабатывается как маркер, и его функция устанавливается как объединение необработанных значений RGB-пикселов. Линейный слой вложения применяется к этому объекту с реальным значением [7], чтобы спроецировать его в произвольное измерение. На этих токенах патча применяется несколько блоков-трансформеров с модифицированным вычислением функции самосознания.

Чтобы выполнить обязанности *«критика»*, использован *механизм оценки* (рис. 7), проверяющий, насколько хорошо выходной midi-файл семантически соответствует входному изображению [11]. С этой целью авторами применён подход *двойного кодировщика*. Модель состоит из двух кодировщиков, работающих с музыкой и изображениями соответственно. *Кодировщик изображения f* берет изображение $x$ и отображает его в вектор $f(x) \in R^d$.

Аналогично, при заданном входном midi-файле $y$ кодировщик музыкальных композиций $g$ извлекает векторное представление $g(y) \in R^d$. Процент совпадения между изображением $x$ и музыкальной композицией $y$ можно вычислить, если оценить сходство векторов $f(x)$ и $g(y)$.

Такие обучающие пары совпадающих изображений и музыки [11], а также параметры управления $f$ и $g$ были оптимизированы так, чтобы сходство этих пар было высоким, а сходство несовпадающих пар – низким. Авторы проводили эксперименты с различными предварительно обученными нейронными сетями *двойных кодировщиков*, но пока не проводили систематического исследования их различий. Этому вопросу будет уделено внимание в дальнейших исследованиях.

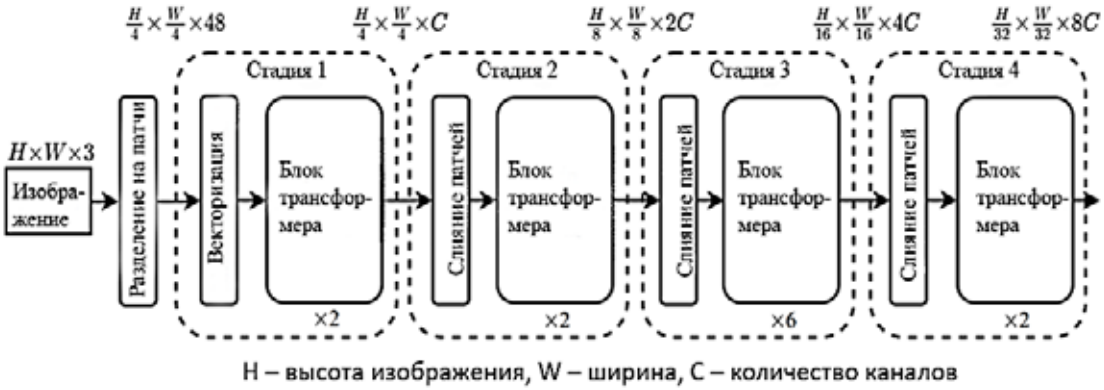



*Рис. 6. Кодировщик изображений*

Для программной реализации поставленной задачи исследования были изучены современные библиотеки и функции программного обеспечения, связанные с разработкой и оценкой качества сгенерированного музыкального ряда. Для этого были импортированы методы и средства библиотек программы Python, такие как music21. Кроме того, для обучения нейронной сети в среде Python, авторами использовался скрипт retrain.py.

**Результаты исследования и их обсуждение**

В этой работе авторы применили иерархическое обучение к модели Transformer для повышения производительности о задачах генерации длинных последовательностей, модифицировав стандартную архитектуру преобразования последовательности в последовательность [12] и добавив механизм иерархического внимания для улучшения обработки длинных последовательностей.

Используется 12 слоев кодировщика и декодировщика скрытого размера 1024×4096, для размеров полностью подключенных сетей прямой связи, и 16 головок внимания, как в кодировщике, так и в декодировщике. В отличие от оригинального алгоритма Transformer, используется активация GELU. Во время предварительной обработки добавляются токены BOS в начало каждого предложения, в каждом исходном документе.

В работе используется тот же кодировщик, что и в алгоритме Transformer Vaswani. Это даёт встраивание для каждого входного токена. После этого добавляется дополнительный слой кодировщика, который касается только вложений токенов BOS, во время предварительной обработки данных. Этот слой авторы назвали *иерархическим кодировщиком*. Слой производит другой уровень контекстных представлений для каждого из токенов BOS. Их можно интерпретировать как представления на уровне предложения. По мнению авторов, один иерархический уровень работает лучше всего, хотя может использоваться и несколько иерархических уровней. Как и в алгоритме Transformer Vaswani, каждый слой декодировщика сначала выполняет *функцию самовнимания* над ранее созданным токеном, а затем следит за выходами финального слоя кодировщика уровня токена. Далее добавляется модуль внимания, который занимается внедрением токенов BOS из слоя иерархического кодировщика (рис. 7).

Стандартный алгоритм Transformer Vaswani использует функцию *глобального самосознания* для установления связи между токеном и всеми другими токенами, что приводит к квадратичному увеличению сложности с размером изображения, что не подходит для интенсивных задач.

Механизм оценки требуется для того, чтобы оценить, насколько хорошо выходной midi- файл семантически соответствует входному изображению. С этой целью авторы выбрали подход двойного кодировщика. Модель *двойного кодировщика* состоит из двух кодировщиков, которые работают с музыкой и изображениями соответственно. Точнее, кодировщик изображения $f$ берет изображение $x$ и отображает его в вектор $f(x)\ Rd$. Аналогично, при заданном входном midi-файле $y$ кодировщик музыки $g$ извлекает соответствующее векторное представление $g(y)\ Rd$. Процент совпадения между образом изображения $x$ и музыкой $y$ можно вычислить, если взять отношения векторов:

$$\frac{f(x)g(y)}{\|f(x)\|^2\|g(y)\|^2}$$

где $||.||^2$ – это квадрат нормы соответствующих векторов.

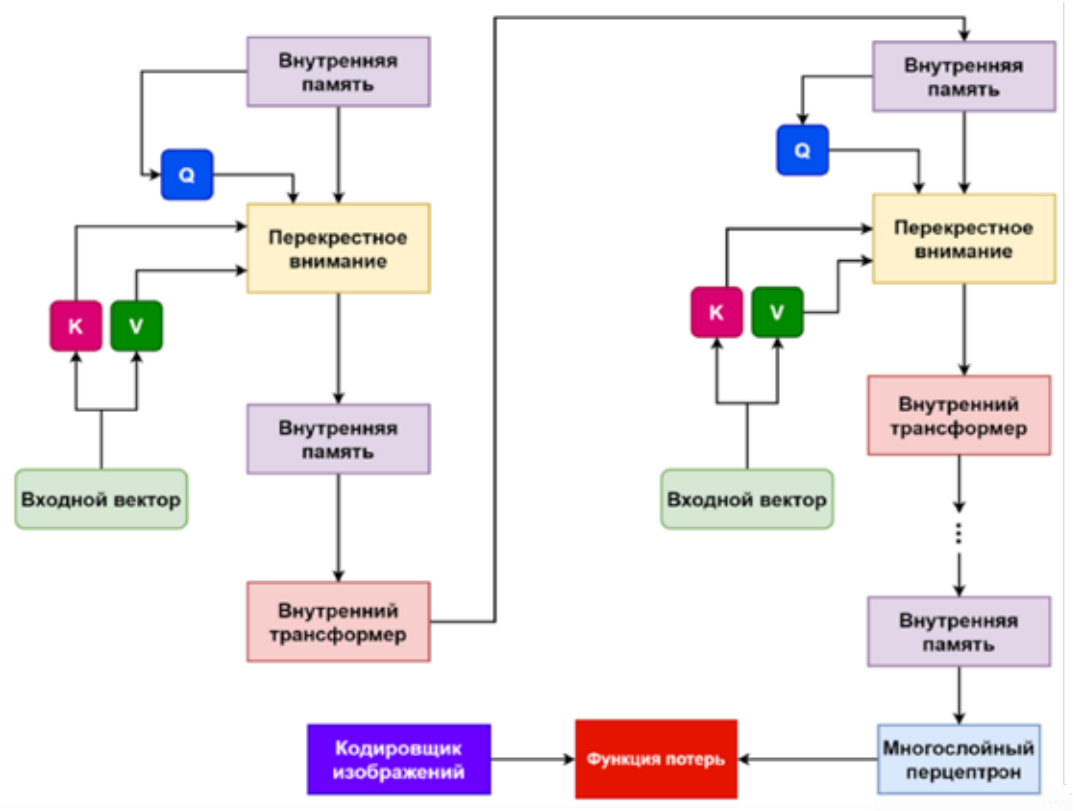



*Рис. 7. Нейронная сеть для оценки качества музыкальной композиции*

Даны обучающие пары совпадающих изображений и музыки, где параметры управления *f* и *g* оптимизированы таким образом, чтобы сходство этих пар было высоким, а сходство генерируемых отрицательных пар – низким. Эксперимент проводился с предварительно обученными двойными кодировщиками, но без систематического исследования их различий.

Результаты разработки программного средства генерации музыкальных композиций *Music Generator–21* (*MG21*) обсуждались в Донском государственном техническом университете г. Ростова-на-Дону. Коллеги провели сравнение представленной разработки с известными зарубежными аналогами программ автоматической генерации музыки *EcrettMusic*, *AlComposer, StyleGAN2* по различным параметрам – по разнообразию музыкальных композиций, качеству музыки и её соответствию изображению, а также оценили новизну разработанного алгоритма.

Музыкальный генератор *MG21* показал достойные результаты: разнообразие генерируемых композиций, при высоком качестве музыки в звуковых midi-файлах и высоком соответствии изображению, передающем настроение, присущее картине, фотографии и даже карикатуре. Отмечено, что полученный midi-файл можно обработать в любом музыкальном редакторе и при желании модифицировать, оркестровать и аранжировать полученную композицию.

## Заключение

Таким образом, поставленная цель исследования достигнута: авторами изучены существующие в мире нейросетевые решения, исследованы основные механизмы, лежащие в основе генеративных моделей и эволюционных алгоритмов нейронных сетей, разработана оригинальная *encoder-decoder* архитектура, применяющая кодировщики изображений, выбран оптимальный фреймворк, библиотеки и технологии для разработки нейросетевой архитектуры программного средства, создан алгоритм обучения нейронной сети. Разработано импортозамещающее программное средство с комплексом модулей на базе искусственных нейронных сетей, предназначенное для генерации музыкальных композиций по заданным параметрам, на основе произвольных изображений – музыкальный генератор *MG21*.

Результаты работы программы в виде музыкальных файлов были продемонстриро-

ваны и получили высокую оценку. Коллеги отметили, что тема исследований представляет большой интерес, и при дальнейшем её развитии область применения программного средства станет достаточно широкой. Алгоритмические композиции (в помощь человеку – композитору, звукорежиссёру или музыкальному оформителю) могут найти своё применение в различных сферах искусства, шоу-бизнеса и маркетинга, в рекламной деятельности и торговле: для написания как серьёзной, так и популярной музыки, саундтреков к кино-, теле- и мультфильмам, оформления театральных постановок и спектаклей, художественных выставок, при озвучивании экспозиций в мультимедийных музеях, на концертах, при звуковом оформлении общественных мероприятий и в других ситуациях, где должна звучать музыка или любое звуковое сопровождение. При этом настроение генерируемой музыки можно задавать с помощью изображений – картин известных художников, фотографий и даже карикатур. Нейронная сеть, обученная с помощью соответствующих наборов данных, образцов изображений в графических файлах, и образцов музыки в звуковых файлах, способна импровизировать в любом стиле, имитируя творческий стиль конкретного композитора.

Во время демонстрации работы программы были прослушаны музыкальные фрагменты в стиле сочинений Иоганна Себастьяна Баха, Вольфганга Амадея Моцарта, Петра Ильича Чайковского, Эдварда Грига, Мориса Равеля, Александра Николаевича Скрябина, Сергея Васильевича Рахманинова и других известных композиторов, как русских, так и зарубежных. Также были сгенерированы импровизации в стиле русских народных песен, джаза, рок-н-ролла, диско, рэпа и других музыкальных направлений. Подчёркнута необходимость продолжить начатое исследование, доведя реализацию идеи до коммерческого уровня.

**Список литературы**

1. Fernandez Jose D., Vico F. AI Methods in Algorithmic Composition: A Comprehensive Survey. Journal of Artificial Intelligence Research. 2013. Vol. 48. P. 513–582. DOI: 10.1613/jair.3908.

2. Lopes Henrique B., Martins Flavio V., Cardoso Rodrigo T. Combining Rules and Proportions: A Multiobjective Approach to Algorithmic Composition. 2017 IEEE Congress on Evolutionary Computation (CEC). Donostia, Spain. P. 2282–2289. DOI: 10.1109/ CEC.2017.7969581.

3. Vico F., Albarracin-Molina D., Manzoni L. Automatic Music Composition with Evolutionary Algorithms: Digging into the Roots of Biological Creativity. Handbook of Artifical Intelligence for Music. Springer Link. 2021. P. 455–483. DOI: 10.1007/ 978–3–030–72116–9_17.

4. Dong H.-W., Chen K., Dubnov Sh., McAuley J., Kirkpatrick T. Multitrack Music Transformer: Learning Long-Term Dependencies in Music with Diverse Instruments. Computer Sciences. ArXiv.2022. abs/2207.06983. P. 1–8. DOI: 10.48550/arXiv.2207.06983.

5. Barina G., Topirceanu A., Udrescu M. MuseNet: Natural Patterns in the Music Artists Industry. 2014. IEEE 9th International Symposium on Applied Computational Intelligence and Informatics (SACI). Timisoara, Romania. P. 317–322. DOI: 10.1109/SACI.2014.6840084.

6. Vaswani A., Shazeer N., Parmar N., etc. Attention is All you Need. 31th International Conference on Neural Information Processing Systems. 2017. P. 6000–6010. DOI: 10.5555/3295222.3295349.

7. Li M., Xu R., Wang Sh., etc. CLIP–Event: Connecting Text and Images with Event Structures. ArXiv. 2022. Abs/2201.05078. DOI: 10.48550/arXiv.2021.05078.

8. Coca A.E., Romero R.A.F., Zhao L. Generation of Composed Musical Structures through Recurrent Neural Networks Based on Chaotic Inspiration. 2011 IEEE International Joint Conference on Neural Networks. San Jose, CA, USA. P. 3220–3226. DOI: 10.1109/IJCNN.2011.6033648.

9. Набор данных MAESTRO. URL: https://magenta.tensorflow.org/datasets/maestro (дата обращения: 12.08.2022).

10. Liu Z., Lin Yu, Cao Yu, etc. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. 2021 IEEE International Conference on Computer Vision (ICCV). Montreal, QC, Canada. P. 9992–10002. DOI: 10/1109/ICCV48922.2021.00986.

11. Jedrzejewska M., Zjawinski A., Stasiak B. Generating Musical Expression on MIDI Music with LSTM Neural Network. 2018 IEEE 11th International Conference on Human System Interaction (HSI). 2018. Gdansk, Poland. P. 132–138. DOI: 10.1109/HSI.2018.8431033.

12. Kozlov V.K., Garifullin M.S. Transformer State diagnosis in optical spectra of transformer oils. Journal of Engineering and Applied Sciences. 2016. Vol. 11. No. 14. P. 3042–3046. DOI 10.3923/jeasci.2016.3042.3046.