

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра математической кибернетики и компьютерных наук

**СОСТАВЛЕНИЕ НОТНЫХ ЗАПИСЕЙ ДЛЯ МУЗЫКАЛЬНЫХ
ПРОИЗВЕДЕНИЙ С ПОМОЩЬЮ НЕЙРОННЫХ СЕТЕЙ**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 411 группы

направления 02.03.02 — Фундаментальная информатика и информационные
технологии

факультета КНиИТ

Катимуллина Айдара Даниярбековича

Научный руководитель

доцент, к. ф.-м. н.

А. И. Иванова

Заведующий кафедрой

к. ф.-м. н., доцент

С. В. Миронов

Саратов 2023

ВВЕДЕНИЕ

В современном мире огромное количество информации представлено в текстовом формате. Информационные системы и социальные медиа платформы ежедневно генерируют огромные объемы текстовых данных, которые необходимо быстро и эффективно обрабатывать. За последние 10 лет технологии глубокого обучения нашли свое повсеместное применение: от анализа больших данных до качественного машинного перевода. Развитие этих технологий позволяют разрабатывать более сложные модели нейросетей и решать такие проблемы, как, например, автоматизация беспилотных транспортных средств, внедрение систем для распознавания лиц.

Составление MIDI-файла из аудиофайла — это процесс создания нотной записи в универсальном формате .mid, воспринимаемым большинством нотных редакторов. Данная задача является сложной, поскольку требует идентификации характерных интервалов и особенностей ритмики в музыкальном произведении, понимания их взаимосвязей и создания корректных нот.

Несмотря на большое кол-во попыток решить проблему, практически применимая, универсальная система музыкальной транскрипции не существует и по сей день. Существующие системы на порядок хуже обученного человека в точности и гибкости. Как ни странно, даже транскрипция одноголосого произведения (т.е. лишь одной только мелодии) является ещё нерешённой проблемой.

Целью данной работы является исследование и разработка алгоритма машинного обучения для автоматического подбора нот с помощью фреймворка глубокого обучения PyTorch.

Для достижения поставленной цели необходимо выполнение следующих задач:

- исследовать существующие методы генерации нот;
- реализовать программный код, позволяющий генерировать нотную запись в виде MIDI-файла для выбранного аудиофайла.

Структура и объем работы

Для решения поставленных задач выполнена выпускная квалификационная работа, включающая в себя введение, 2 основные главы, заключение, список использованных источников из 24 наименований и одного приложения. Работа изложена на 62 страницах и содержит 22 рисунка.

Первая глава имеет название «Теоретические основы нейронных сетей» и содержит основную информацию о нейронных сетях: определения, виды, архитектуры.

Вторая глава имеет название «Реализация приложения», данная глава содержит описание процесса выполнения работы.

Выпускная квалификационная работа заканчивается заключением, списком использованных источников, а также приложением с кодом А.

Основное содержание работы

Первый раздел содержит необходимые теоретические сведения, касающиеся теории, и принципы работы нейронных сетей. Приводятся основные определения, используемые в дипломной работе.

Искусственная нейронная сеть (ИНС) (или же просто нейронная сеть (НС)) — упрощенная модель нервной системы, т.е. биологической сети. Она представляет собой набор искусственных нейронов, которые взаимодействуют друг с другом. Базовые принципы их работы были описаны Уорреном МакКаллоком и Уолтером Питтсом в 1943 году, а в 1957 году нейрофизиолог Фрэнк Розенблатт создал первую нейронную сеть.

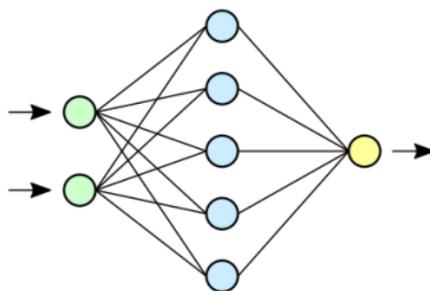


Рисунок 1 – Схема простой нейронной сети

На рисунке 1 представлена схема простой искусственно нейронной сети. Входные нейроны помечены зелёным, голубым — скрытые нейроны, выходной нейрон — жёлтым. Так, например, искусственный нейрон, из которых состоит ИНС, имеет намного более простую структуру: у него есть несколько входов, на которых он принимает различные сигналы, преобразует их и передает другим нейронам, таким образом, он преобразует n входных параметров в один выходной.

Свёрточная нейронная сеть (англ. convolutional neural network, CNN) — специальная архитектура искусственных нейронных сетей, предложенная Яном Лекуном в 1988 году и нацеленная на эффективное распознавание образов, входит в состав технологий глубокого обучения (англ. deep learning). Она пользуется некоторыми особенностями зрительной коры, в которой были открыты так называемые простые клетки, реагирующие на прямые линии под разными углами, и сложные клетки, реакция которых связана с активацией определённого набора простых клеток. Таким образом, идея свёрточных нейронных сетей заключается в чередовании свёрточных слоёв (англ. convolution layers) и субдис-

кретизирующих слоёв (англ. subsampling layers или pooling layers). Структура сети — однонаправленная (без обратных связей), многослойная. Для обучения используются стандартные методы, чаще всего метод обратного распространения ошибки. Функция активации нейронов — любая, по выбору исследователя.

Название архитектура сети получила из-за наличия операции свёртки, суть которой в том, что каждый фрагмент изображения умножается на матрицу (ядро) свёртки поэлементно, а результат суммируется и записывается в аналогичную позицию выходного изображения.

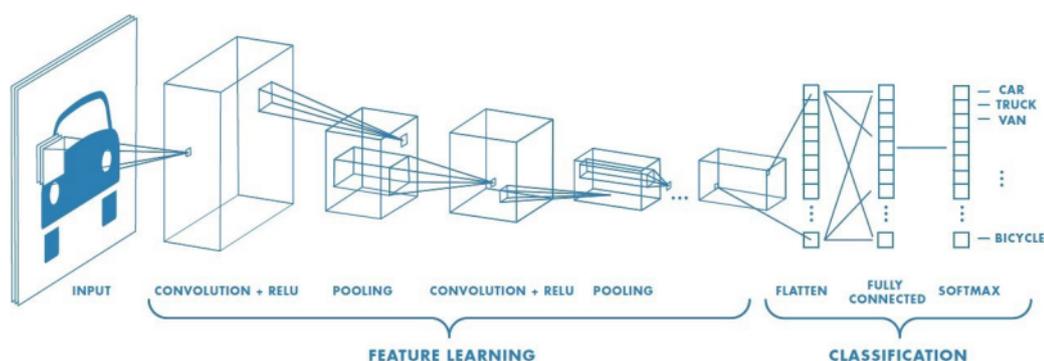


Рисунок 2 – Архитектура сверточной нейронной сети

Рекуррентные нейронные сети (РНС, Recurrent Neural Network, RNN) — вид нейронных сетей, где связи между элементами образуют направленную последовательность. Эта структура предоставляет возможность обрабатывать последовательности, будь то пространственные цепочки или же какие-то серии событий. Общая схема рекуррентной сети представлена на рисунке 3, а на рисунке 4 представлен слой сети. Главным отличием от многослойных перцептронов является то, что RNN способны свою внутреннюю память использовать для обработки произвольных по длине цепочек. Это позволяет применять рекуррентные сети в таких задачах, где различные части вместе представляют что-то единое, например, обработка естественного языка. Для рекуррентных сетей было предложено большое количество архитектур, как сложных, так и весьма простых.

Благодаря этому появляется возможность обрабатывать серии событий во времени или последовательные пространственные цепочки. В отличие от многослойных перцептронов, рекуррентные сети могут использовать свою внутреннюю память для обработки последовательностей произвольной длины. Поэтому сети RNN применимы в таких задачах, где нечто целостное разбито на части,

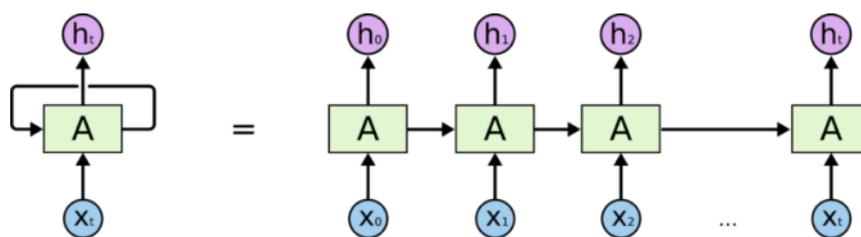


Рисунок 3 – Рекуррентная нейронная сеть

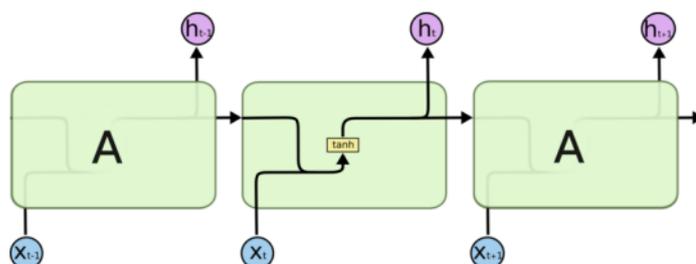


Рисунок 4 – Схема слоя рекуррентной сети

например: распознавание рукописного текста или распознавание речи. Было предложено много различных архитектурных решений для рекуррентных сетей от простых до сложных. В последнее время наибольшее распространение получили сеть с долговременной и кратковременной памятью (LSTM) и управляемый рекуррентный блок (GRU).

Во втором разделе приводится описание реализованной в ходе выполнения дипломной работы программы, состоящей из двух подпрограмм, одна из которых реализует веб-сервис, а вторая отвечает за обучение и вывод описания.

Язык программирования Python. Python — высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ. Python стал одним из самых популярных языков, он используется в анализе данных, машинном обучении, DevOps и веб-разработке, а также в других сферах, включая разработку игр.

Docker. Docker — это платформа для разработки, доставки и запуска контейнерных приложений. Docker позволяет создавать контейнеры, автоматизировать их запуск и развертывание, управляет жизненным циклом. Он позволяет запускать множество контейнеров на одной хост-машине.

Контейнеры Docker являются изолированными средами выполнения, которые содержат все необходимые для работы приложения файлы, библиотеки и зависимости. Они могут быть созданы и развернуты на любом компьютере, который поддерживает Docker, без необходимости установки всех зависимостей и конфигурации окружения.

Pytorch. PyTorch, открытый исходный код глубокого обучения, предлагает множество преимуществ:

1. **Динамический вычислительный граф:** Одним из главных преимуществ PyTorch является его поддержка динамического вычислительного графа, который позволяет более гибко определять модели и легче отлаживать код. Этот подход делает PyTorch более интуитивно-понятным для многих разработчиков и обеспечивает большую гибкость.
2. **Простота в использовании:** PyTorch предлагает простой в использовании Python API, что делает его привлекательным для начинающих и опытных разработчиков. API PyTorch часто описывается как более интуитивный и близкий к стандартному Python, что ускоряет процесс обучения и разработки.
3. **Сильная поддержка сообщества и активное развитие:** PyTorch имеет огромное сообщество, активно внедряющее новые функции, модели и методы обучения. Сообщество также создает множество библиотек и инструмен-

тов, расширяющих возможности PyTorch для решения разнообразных задач глубокого обучения.

4. Отличная поддержка для работы с GPU: PyTorch обеспечивает простую и эффективную работу с графическими процессорами, что делает его отличным выбором для задач, требующих высокой вычислительной мощности, таких как обучение глубоких нейронных сетей.
5. Использование в индустрии и академических исследованиях: PyTorch широко применяется в индустрии и академических исследованиях благодаря своей гибкости, производительности и обширным возможностям для решения различных задач машинного обучения и глубокого обучения.
6. Непрерывное развитие и улучшение: Команда разработчиков постоянно улучшает PyTorch, предлагая новейшие инструменты, методы и модели, делая его одним из наиболее актуальных фреймворков для глубокого обучения.

Использование PyTorch обычно рекомендуется в задачах, связанных с исследованиями и разработкой глубоких нейронных сетей, поскольку он предоставляет мощный и гибкий инструментарий для работы с данными и создания сложных моделей глубокого обучения

Реализация приложения. В ходе работы создано приложения для составления нотных записей из аудиофайла. Интерфейс приложения представлено в виде веб-интерфейса. В работе используется один датасет. Он содержит аудиофайлы, информацию о них и соответствующий им MIDI-файл. Для создания приложения используется язык программирования Python версии 3.11, для управления Python-модулями также использован инструмент Poetry, для развёртывания приложения используется инструмент Docker.

Оценка рекомендательной системы. Для оценки реализованной рекомендательной системы была отобрана выборка из аудиофайлов, не участвовавших в обучении модели. В качестве метрики точности была использована метрика F-score. Она помогает учитывать точность, когда классы не сбалансированы, сосредоточив внимание на точности положительных прогнозов и фактически положительных записей.

$Precision = TP / (TP + FP)$, показывает долю ответов, которые действительно являются положительными и которые программа отметила положительными.

Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

Рисунок 5 – Confusion-матрица

$Recall = TP / (TP + FN)$, показывает, сколько положительных ответов из их общего количества было отмечены программой.

Простейшая метрика F-score имеет следующую формулу: $F1 = 2 * (Precision * Recall) / (Precision + Recall)$.

Оценка точности модели дала следующие результаты:

- 76.63 для Precision
- 70.12 для Recall
- 73.11 для F1

ЗАКЛЮЧЕНИЕ

В ходе выполнения работы были рассмотрены различные виды нейронных сетей и методов работы с изображениями и текстов, такие как:

- сверточные нейронные сети;
- рекуррентные нейронные сети;
- энкодер-декодер.

В практической части работы реализована программа, позволяющая обучить нейронную сеть для генерации MIDI-файла из конкретного исполнения музыкального произведения.

В ходе решения задачи реализации пользовательского интерфейса были получены навыки разработки FastAPI.

В ходе решения задачи реализации генерации MIDI-файлов были проанализированы некоторые виды алгоритмов, а также способы их оценки. Точность по метрике F1 составила 73 процента.

Таким образом, поставленные задачи решены в полном объеме и достигнута поставленная цель — создано приложение, способное с использованием нейронных сетей создать нотную запись из аудиофайла.