



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Институт информационных технологий (ИИТ)
Кафедра прикладной математики (ПМ)

Утверждаю

И.о. заведующего кафедрой ПМ

 Смоленцева Т.Е.
(подпись)

«01» марта 2024 г.

ЗАДАНИЕ

на выполнение курсовой работы
по дисциплине «Методы анализа данных»

Студент Ковалев Алексей Леонидович

Группа ИММО-01-23

Тема «Метод определения настроения на основе математических алгоритмов и моделей»

Исходные данные: выбранные студентом компании

Перечень вопросов, подлежащих разработке, и обязательного графического материала:

Характеристика деятельности организации и общее описание

Структура исследуемых акций и графические модели процессов (исходный масштаб, логарифмический масштаб, анализ темпов роста системы)

Анализ акций на принадлежность к процессам интенсивного роста, графическое представление (выявление участков интенсивного роста, анализ выявленных участков)

Анализ данных на периодичность, графическое представление (исключение тренда, построение сдвиговой функции)

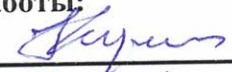
Исследование данных на принадлежность процесса к моделям ограниченного роста, графическое представление (рассмотреть модель Гомперца, логистическую модель)

Применение ячейки развития для анализа критических уровней, графическое представление (построение ячейки для развития для уровня T_k , прогнозирование T_{k+1})

Срок представления к защите курсовой работы:

до «30» мая 2024 г.

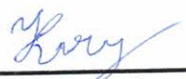
Задание на курсовую работу выдал


Подпись руководителя

Кузьмин В.И.
(ФИО руководителя)

«01» марта 2024 г.

Задание на курсовую работу получил


Подпись обучающегося

Ковалев А.Л.
(ФИО обучающегося)

«01» марта 2024 г.



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Институт информационных технологий (ИИТ)
Кафедра прикладной математики (ПМ)

КУРСОВАЯ РАБОТА

по дисциплине «Методы анализа данных»

Тема курсовой работы: «Метод определения настроения на основе математических алгоритмов и моделей»

Студент группы ИММО-01-23 Ковалев Алексей Леонидович


(подпись)

Руководитель
курсовой работы

профессор, д.т.н., Кузьмин В.И.


(подпись)

Работа представлена к защите «__»_____2024 г.

Допущен к защите «__»_____2024 г.

Москва 2024 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Исходные данные	6
2 Выбор аудио признаков	8
3 Обучение модели	18
4 Выводы о работе	25
Заключение	27
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ	28

ВВЕДЕНИЕ

Актуальность и цель исследования

В современном мире эмоции играют ключевую роль в человеческом взаимодействии и коммуникации. С развитием технологий и увеличением объема данных, доступных для анализа, стало возможным автоматическое распознавание эмоций на основе различных видов данных, таких как текст, видео и аудио. Одной из наиболее интересных и перспективных областей является анализ аудио признаков для предсказания настроений, что открывает новые возможности для приложений в области медицины, маркетинга, развлечений и многих других.

Цель работы

Цель данной курсовой работы заключается в разработке нейронной сети для предсказания настроений на основе аудио признаков. Мы стремимся создать модель, которая будет способна эффективно классифицировать эмоции, выраженные в аудио файлах, используя современные методы машинного обучения и глубокого обучения.

Задачи исследования

1. Сбор и подготовка данных: создание набора данных аудио файлов с разметкой эмоций и извлечение релевантных аудио признаков.
2. Разработка базовой модели: создание нейронной сети с использованием стандартных архитектур и методов.
3. Оптимизация и улучшение модели: применение различных методов регуляризации, изменения архитектуры сети, гиперпараметрической оптимизации и других техник для повышения точности предсказаний.
4. Оценка и анализ результатов: проведение экспериментов, оценка производительности модели на тестовых данных и анализ полученных результатов.

Научная новизна

Научная новизна данной работы заключается в применении современных методов глубокого обучения для задачи предсказания настроений на основе аудио признаков, а также в использовании различных техник оптимизации и регуляризации для достижения высоких показателей точности. Исследование проводилось с целью создания модели, которая

может быть использована в реальных приложениях, требующих автоматического распознавания эмоций.

1 Исходные данные

1.1 Выбор исходных данных

Датасет, используемый в данной курсовой работе, состоит из 79088 аудиофайлов в формате WAV, каждый из которых размечен по одному из четырех эмоциональных состояний: позитив, нейтраль, грусть и злость/раздражение. Аудиозаписи в данных содержат фразы, произнесенные разными людьми, с четкими инструкциями относительно того, как должны быть выражены эмоции.

Категории эмоций

1. Позитив: Фразы произносятся с улыбкой или смехом, с выраженными ударениями на словах, подчеркивающих позитивное настроение.
2. Нейтраль: Фразы читаются максимально «плоско», без выражения каких-либо эмоций.
3. Грусть: Текст произносится с грустью, как будто у говорящего был тяжелый день или он испытывает личные проблемы.
4. Злость/раздражение: Фразы произносятся со злостью, с элементами крика и негодования

Процесс создания датасета

Этап 1: Разметка первой группы

Изначально тексты были озвучены первой группой разметчиков, которым были даны четкие инструкции по выражению каждой эмоции. Эта часть работы была выполнена на краудсорсинговых платформах, где разметчики произносили тексты согласно указанным эмоциям.

Этап 2: Проверка второй группой

Для обеспечения качества разметки была привлечена вторая группа разметчиков, которая прослушивала записи и определяла эмоции без учета текста. В датасет вошли только те записи, в которых услышанная эмоция совпадала с заданной первой группой разметчиков. Этот процесс проверки гарантировал высокую точность разметки эмоциональных состояний в аудиозаписях. Эту часть датасета называли Crowd.

Этап 3: Добавление реальных эмоций

Понимая, что живые эмоции могут несколько отличаться от озвученных, в датасет была добавлена вторая часть, содержащая реальные эмоции в разговорной речи. Для этого были собраны короткие нарезки русскоязычных подкастов, содержащие до пяти слов. Аудиозаписи делались на профессиональные микрофоны, что добавило разнообразия в набор данных. Пропорции между различными эмоциями в этом сегменте данных сохранялись, с преобладанием нейтральных эмоций.

Технические детали данных

1. Источник данных: краудсорсинговые платформы и русскоязычные подкасты.
2. Общее количество записей: 79088.
3. Формат аудиофайлов: WAV.
4. Частота дискретизации: Все аудиозаписи были унифицированы до частоты 16KHz, что является наиболее распространенным качеством и обеспечивает однородность данных.
5. Продолжительность записей: Каждая запись содержит до пяти слов

2 Выбор аудио признаков

2.1 MFCC

Мел-частотные кепстральные коэффициенты (MFCC) являются одними из самых распространенных признаков, используемых в задачах обработки аудио и распознавания речи. MFCC преобразуют аудиосигнал в набор значений, которые лучше соответствуют тому, как человеческое ухо воспринимает звук. Рассмотрим подробнее, что представляют собой MFCC и как они вычисляются.

Мел-частотные кепстральные коэффициенты (MFCC) - это набор коэффициентов, которые представляют короткие амплитудные спектры звука на нелинейной мел-шкале частот. Эта шкала примерно соответствует восприятию частоты человеческим ухом, делая MFCC особенно полезными для задач распознавания речи и анализа аудио.

Вычисление MFCC

Процесс вычисления MFCC состоит из нескольких этапов:

1. Разбиение сигнала на фреймы (окна):

Аудиосигнал разбивается на короткие участки (фреймы) длительностью 20-40 мс. Это делается для того, чтобы предположить, что аудиосигнал в каждом фрейме стационарен.

2. Применение окна Ханнинга (Hanning window):

К каждому фрейму применяется окно Ханнинга для уменьшения эффекта разрыва в начале и конце фрейма.

3. Преобразование Фурье (FFT):

Для каждого фрейма выполняется быстрое преобразование Фурье (FFT), чтобы получить спектральное представление сигнала.

4. Применение мел-фильтров (Mel filter bank):

Спектр преобразуется в мел-шкалу частот с помощью набора треугольных фильтров, равномерно распределенных по мел-шкале. Это преобразование приближает частотную шкалу к восприятию человеческого уха.

5. Логарифм амплитуд (Logarithm of amplitudes):

Амплитуды, полученные после применения мел-фильтров, подвергаются логарифмическому преобразованию.

6. Обратное преобразование Фурье (DCT):

Применяется дискретное косинусное преобразование (DCT) для получения мел-частотных кепстральных коэффициентов. Обычно выбирается несколько первых коэффициентов, так как они содержат основную информацию о спектре.

Преимущества MFCC

1. Близость к восприятию человеческого уха:

Мел-шкала частот, используемая в MFCC, отражает нелинейное восприятие частот человеком, что делает эти коэффициенты особенно полезными для задач распознавания речи и звуков.

2. Эффективное представление аудиосигнала:

MFCC конденсируют информацию о спектре аудиосигнала в небольшое количество коэффициентов, что делает их удобными для использования в моделях машинного обучения.

3. Инвариантность к изменениям громкости:

Логарифмическое преобразование амплитуд делает MFCC менее чувствительными к изменениям громкости, что способствует устойчивости признаков.

Применение MFCC

1. Распознавание речи:

MFCC являются основным признаком в системах автоматического распознавания речи (ASR), так как они эффективно представляют звуковую информацию.

2. Анализ эмоций в речи:

Из-за своей способности улавливать нюансы звукового сигнала, MFCC часто используются в задачах анализа эмоций в речи.

3. Классификация звуков и музыки:

MFCC применяются для классификации различных типов звуков и музыкальных жанров, благодаря их способности выделять важные акустические характеристики.

2.2 Спектральный центроид

Спектральный центроид - это мера, которая указывает на "центр масс" спектра звука. Он вычисляется как взвешенная средняя частота всех компонент спектра, где весами выступают амплитуды соответствующих частотных компонент. Спектральный центроид часто используется для характеристики тембра звука и может быть связан с восприятием яркости или остроты звука.

Вычисление спектрального центроида

Процесс вычисления спектрального центроида включает следующие этапы:

1. Разбиение сигнала на фреймы (окна):

Аудиосигнал разбивается на короткие фреймы длительностью 20-40 мс. Это делается для того, чтобы предположить, что аудиосигнал в каждом фрейме стационарен.

2. Применение окна Ханнинга (Hanning window):

К каждому фрейму применяется окно Ханнинга для уменьшения эффекта разрыва в начале и конце фрейма.

3. Преобразование Фурье (FFT):

Для каждого фрейма выполняется быстрое преобразование Фурье (FFT), чтобы получить спектральное представление сигнала.

4. Вычисление спектрального центроида:

Спектральный центроид для каждого фрейма вычисляется по формуле (2.2):

$$Spectral\ Centroid = \frac{\sum_{k=0}^{N-1} f(k) * S(k)}{\sum_{k=0}^{N-1} S(k)} \quad (2.2)$$

Где:

$f(k)$ - частота k -того бинта,

$S(k)$ - амплитуда в k -том бинте,

N - количество бинтов.

Преимущества спектрального центроида

1. Информативность:

Спектральный центроид предоставляет информацию о расположении "массы" частот в спектре, что полезно для анализа тембра и характера звука.

2. Простота вычисления:

Спектральный центроид легко вычисляется и требует минимальной обработки данных.

3. Применимость к различным задачам:

Этот признак используется в различных областях, включая классификацию звуков, анализ музыки и распознавание речи.

Применение спектрального центроида

1. Анализ и классификация музыки:

Спектральный центроид помогает различать музыкальные инструменты и жанры музыки, поскольку он отражает яркость и тембр звука.

2. Обработка речи:

В задачах распознавания речи спектральный центроид используется для различения фонем и анализа качества речи.

3. Обнаружение событий:

В задачах обнаружения звуковых событий спектральный центроид используется для определения изменений в спектральном содержимом, что помогает в идентификации различных звуков.

2.3 Chroma STFT

Хрома-функция кратковременного преобразования Фурье (Chroma STFT) — это метод представления звука, который показывает интенсивность различных музыкальных нот (хроматических нот) в аудиосигнале. Хроматическая шкала состоит из 12 нот (C, C#, D, D#, E, F, F#, G, G#, A, A#,

B), и Chroma STFT анализирует, какие из этих нот присутствуют в аудиосигнале и с какой интенсивностью.

Вычисление Chroma STFT

Процесс вычисления Chroma STFT состоит из следующих шагов:

1. Разбиение сигнала на фреймы (окна):

Аудиосигнал разбивается на короткие участки (фреймы), чтобы предположить, что сигнал в каждом фрейме стационарен.

2. Применение окна Ханнинга (Hanning window):

К каждому фрейму применяется окно Ханнинга для уменьшения эффекта разрыва в начале и конце фрейма.

3. Быстрое преобразование Фурье (FFT):

Для каждого фрейма выполняется быстрое преобразование Фурье (FFT), чтобы получить спектр.

4. Преобразование частот в хроматические ноты:

Частоты преобразуются в хроматические ноты (C, C#, D, D#, E, F, F#, G, G#, A, A#, B) с помощью функции **librosa.feature.chroma_stft**.

librosa.feature.chroma_stft - это функция из библиотеки **librosa**, которая используется для извлечения хроматических признаков из аудиосигнала с использованием коротковременного преобразования Фурье (STFT). Хроматические признаки представляют собой распределение энергии звука по полутонам (то есть по цветам в музыкальной шкале), что позволяет описать музыкальные характеристики аудиосигнала.

Процесс работы **librosa.feature.chroma_stft**:

1. STFT (Коротковременное преобразование Фурье): Сначала аудиосигнал разбивается на короткие участки времени. Для каждого из этих участков вычисляется преобразование Фурье, что позволяет перевести сигнал из временной области в частотную.

2. Хроматический анализ: Для каждого участка времени вычисляется энергия звука в каждом полутоне музыкальной шкалы. Это делается путем присвоения каждому частотному бину в STFT соответствующего полутонного класса. Например, если у нас есть 12 полутонных классов (C, C#, D, D#, и так далее), то для каждого участка времени будет вычислено распределение энергии по этим классам.

3. Нормализация: После вычисления хроматических признаков они часто нормализуются для уменьшения влияния громкости аудиосигнала.

4. Окончательный результат: Функция возвращает хроматические признаки в виде матрицы, где строки представляют собой различные полутоновые классы, а столбцы представляют собой различные участки времени.

Преимущества Chroma STFT

1. Анализ гармонической структуры:

Chroma STFT позволяет анализировать гармоническую структуру музыки, выявляя присутствие и интенсивность различных нот.

2. Идентификация музыкальных аккордов и мелодий:

Этот признак полезен для идентификации музыкальных аккордов и мелодий, так как он показывает, какие ноты играют и с какой интенсивностью.

3. Распознавание музыки и музыкальной информации:

Chroma STFT часто используется в задачах распознавания музыки и музыкальной информации, таких как определение тональности, жанра и идентификация музыкальных фрагментов.

Применение Chroma STFT

1. Анализ музыкальной информации:

Chroma STFT широко используется для анализа музыкальной информации, помогая определять аккорды, мелодии и гармонии в музыкальных произведениях.

2. Распознавание и классификация музыки:

Этот признак используется в системах распознавания музыки и классификации жанров, так как он эффективно выделяет важные музыкальные характеристики.

3. Тональная идентификация:

Chroma STFT помогает в задачах тональной идентификации, таких как определение тональности музыкального фрагмента или нахождение соответствия между музыкальными фрагментами.

2.4 RMSE

RMS амплитуды (иногда называемая просто RMS) является мерой энергии сигнала и используется для оценки громкости или интенсивности аудиосигнала. Она рассчитывается как квадратный корень из среднего квадратичного значения амплитуды сигнала.

Расчет RMS амплитуды

Для последовательности амплитуд y_i RMS вычисляется по формуле:

$$RMS = \sqrt{\frac{1}{n} \sum_{i=1}^n y_i^2} \quad (2.4)$$

Где:

n - количество выборок в сигнале,

y_i - значение амплитуды на i -той выборке.

Интерпретация RMS амплитуды

- RMS амплитуда предоставляет информацию о мощности сигнала. Высокие значения RMS указывают на более громкий сигнал, в то время как низкие значения указывают на более тихий сигнал.
- RMS амплитуда часто используется в задачах, связанных с анализом динамики аудиосигналов, таких как обнаружение ударов, оценка громкости и динамическая компрессия.

Преимущества RMS амплитуды

1. Мера энергии сигнала:

RMS амплитуда отражает мощность сигнала, что позволяет оценить общую энергию звукового сигнала. Это полезно для анализа громкости и динамики аудио.

2. Инвариантность к знаку амплитуды:

Поскольку амплитуды возводятся в квадрат перед усреднением, RMS амплитуда не зависит от того, положительные или отрицательные значения имеет сигнал, что позволяет получить более стабильные и надежные оценки мощности сигнала.

3. Чувствительность к изменениям амплитуды:

RMS амплитуда чувствительна к изменениям в амплитуде сигнала, что делает её полезной для детектирования изменений громкости и динамических переходов в аудио.

4. Сравнительность:

RMS амплитуда позволяет сравнивать различные сигналы по их энергетическим характеристикам, что полезно в задачах классификации и анализа аудио.

Применение RMS амплитуды

1. Анализ громкости:

RMS амплитуда часто используется для оценки громкости звукового сигнала. В музыкальной обработке и микшировании RMS амплитуда помогает установить уровни громкости различных треков.

2. Обнаружение событий:

В задачах обработки речи и аудио RMS амплитуда используется для обнаружения событий, таких как начало и конец слов, ударов или других кратковременных звуковых событий.

3. Контроль динамики:

В аудиоинженерии RMS амплитуда применяется для настройки динамических процессоров, таких как компрессоры и лимитеры, которые регулируют динамический диапазон сигнала.

4. Классификация аудио:

В задачах классификации звуков, таких как распознавание речи, музыкальных жанров или звуков природы, RMS амплитуда может служить одним из признаков для машинного обучения.

5. Оценка качества звука:

RMS амплитуда используется для оценки качества звука в аудиосистемах и при кодировании аудио. Она помогает выявить изменения в мощности сигнала, которые могут указывать на искажения или шум.

6. Медицинские приложения:

В медицинских приложениях, таких как анализ ЭЭГ или ЭКГ, RMS амплитуда помогает оценивать энергетические характеристики сигналов, что может быть полезно для диагностики.

2.5 Спектральный контраст

Спектральный контраст (Spectral Contrast) - это аудиофича, которая измеряет различие в амплитуде между пиками и впадинами в спектре аудиосигнала. Этот параметр позволяет оценить текстурные и тимбральные характеристики звука и широко используется в задачах классификации и анализа аудио.

Как работает спектральный контраст

1. Преобразование Фурье:

Сначала аудиосигнал преобразуется в частотную область с помощью коротковременного преобразования Фурье (STFT).

2. Разделение на полосы частот:

Частотный спектр разбивается на несколько полос частот (обычно логарифмически), и для каждой полосы вычисляются пики (максимумы) и впадины (минимумы).

3. Вычисление контраста:

Спектральный контраст для каждой полосы определяется как разность в дБ между средними значениями амплитуды пиков и впадин.

Преимущества спектрального контраста

1. Характеризация тимбра:

Спектральный контраст эффективно описывает тимбральные характеристики звука, помогая различать звуки с разной текстурой и гармонической структурой.

2. Инвариантность к громкости:

Поскольку контраст вычисляется на основе относительных уровней пиков и впадин, он менее подвержен влиянию общей громкости сигнала, чем абсолютные амплитуды.

3. Выявление структуры:

Спектральный контраст может выявлять различия в спектральной структуре между разными типами звуков, что полезно для задач классификации аудио и распознавания образов.

Применение спектрального контраста

1. Классификация музыкальных жанров:

Спектральный контраст помогает различать музыкальные жанры, так как разные жанры часто имеют характерные тимбральные особенности.

2. Распознавание речи и спикеров:

В задачах распознавания речи и идентификации говорящих спектральный контраст может быть использован для выявления уникальных тимбральных признаков голоса.

3. Анализ акустической среды:

Спектральный контраст применяется для анализа акустических характеристик помещений и окружающей среды, помогая различать разные типы звуковых источников.

4. Музыкальная информационная ретривация (MIR):

В области MIR спектральный контраст используется для извлечения и анализа аудиофич, что помогает в задачах поиска и классификации музыкальных композиций.

3 Обучение модели

3.1 Выбор модели

Задача подразумевает определение одной из нескольких возможных категорий эмоций (например, счастье, грусть, гнев и т.д.) на основе входных данных. Поскольку результатом является одна из заранее определённых категорий, задача является классической задачей классификации.

Выбор классификационной модели

Подходящая архитектура: Классификационные модели, особенно нейронные сети, хорошо справляются с задачами, где необходимо научиться различать и распознавать шаблоны в данных. Для задачи распознавания эмоций важно, чтобы модель могла выявлять тонкие различия в характеристиках, которые соответствуют разным эмоциям.

Альтернативные подходы и их ограничения

1. **Регрессия:** Задачи регрессии нацелены на предсказание непрерывных значений. Однако, в контексте распознавания эмоций, результат должен быть категорией, а не непрерывным значением. Применение регрессии потребовало бы дополнительных шагов для преобразования непрерывных значений в категории, что усложняет и не улучшает решение задачи.

2. **Кластеризация:** Хотя кластеризация полезна для группировки данных на основе сходства, она не предназначена для использования в задачах с известными метками. В нашей задаче мы имеем заранее определённые категории эмоций, которые мы хотим предсказать. Кластеризация также не обеспечивает возможность оценивать точность модели, что важно для отслеживания и улучшения её производительности.

3. **Генеративные модели:** Такие модели, как вариационные автоэнкодеры (VAEs) или генеративно-состязательные сети (GANs), полезны для генерации новых данных или изучения скрытых представлений. Однако они не предназначены для непосредственного решения задач классификации, где необходимо предсказать конкретные метки классов на основе входных данных.

Преимущества классификационной модели

Интерпретируемость результатов: Классификационные модели предоставляют прямой способ предсказания категории для каждого входного примера. Это делает результаты легко интерпретируемыми и понятными.

Эффективность обучения: Классификационные модели, такие как нейронные сети, могут эффективно обучаться на большом количестве данных с метками, что позволяет достигать высокой точности и обобщающей способности.

Гибкость и адаптивность: Модели классификации можно легко адаптировать и настраивать под конкретные задачи, используя различные архитектуры нейронных сетей и методы регуляризации для улучшения производительности.

Модель классификации является естественным и наиболее подходящим выбором для задачи распознавания эмоций. Она предоставляет прямой и эффективный способ предсказания категорий эмоций на основе входных данных, что обеспечивает высокую точность, интерпретируемость и адаптивность к изменениям и улучшениям в данных и модели.

Модель **ImprovedEmotionClassifier** была выбрана и сконструирована для решения задачи классификации эмоций по нескольким причинам:

1. Архитектура сети

- **Многослойная структура:** Модель состоит из нескольких полносвязных слоев, что позволяет ей захватывать сложные нелинейные зависимости в данных.
- **Размерность скрытых слоев:** Выбор размера скрытых слоев (256 нейронов) основан на необходимости уравновесить способность модели к обучению сложных представлений и предотвращение переобучения. Большое количество нейронов в скрытых слоях позволяет модели эффективно обучаться, а также увеличивает ее выразительную мощность.
- **Dropout:** Использование Dropout слоев с вероятностью 0.3 помогает уменьшить вероятность переобучения, так как он случайным образом отключает нейроны во время обучения, что заставляет модель обучаться более устойчивым и обобщающим признакам.

2. Оптимизация и регулирование

- **Adam Optimizer:** Выбор оптимизатора Adam обусловлен его способностью эффективно и быстро сходиться за счет адаптивного изменения шага обучения для каждого параметра. Он хорошо работает на практике для большинства задач машинного обучения.
- **CrossEntropyLoss:** Потери перекрестной энтропии являются стандартным выбором для задач многоклассовой классификации, так как они

измеряют разницу между предсказанными и реальными вероятностями классов.

- **ReduceLROnPlateau Scheduler:** Этот планировщик снижает скорость обучения, когда показатель валидационной потери перестает улучшаться, что помогает модели лучше адаптироваться на поздних стадиях обучения и избегать переобучения.

3. Стратегия обучения

- **Ранняя остановка:** Эта техника позволяет прекратить обучение, если точность модели на валидационном наборе перестает улучшаться в течение нескольких эпох. Это помогает предотвратить переобучение и сохранить вычислительные ресурсы.
- **Тщательное тестирование:** После каждой эпохи выполняется оценка на тестовом наборе данных для мониторинга точности и валидационных потерь, что позволяет своевременно вносить коррективы в процесс обучения.

Эти элементы в совокупности позволяют **ImprovedEmotionClassifier** эффективно обучаться и обобщаться на данных задачи классификации эмоций. Модель имеет достаточную гибкость и мощность, чтобы захватывать сложные зависимости в данных, а также включает механизмы для предотвращения переобучения и улучшения обобщающей способности.

3.2 Оценка обучения модели

Точность и потери на тестовом наборе

- **Точность:** Модель достигла точности на тестовом наборе данных 78.24%. Это указывает на то, что модель способна правильно предсказывать эмоции в большинстве случаев.
- **Потери:** Значения потерь на валидационном наборе данных (Val Loss) постепенно снижались, что свидетельствует о том, что модель успешно обучалась и улучшала свои предсказания. Финальное значение потерь составляет около 0.5960, что является хорошим показателем.

Анализ результатов предсказаний

- Модель предсказала большинство примеров как "neutral". Это может указывать на наличие дисбаланса в данных, где категория "neutral" представлена в большем количестве, либо на трудности модели в различении других эмоций.
- Из 10 примеров, приведенных в конце, только один был неправильно классифицирован. Остальные примеры были

классифицированы как "neutral", что также может свидетельствовать о предвзятости модели к этой категории.

Возможные улучшения

- Увеличение разнообразия данных: Если данные имеют сильный дисбаланс классов, то необходимо провести балансировку классов либо с помощью увеличения количества примеров недостаточно представленных классов, либо с помощью методов увеличения данных.
- Улучшение модели: Можно попробовать использовать более сложные архитектуры нейронных сетей, например, добавить дополнительные слои или изменить гиперпараметры модели.
- Регуляризация: Хотя Dropout помогает уменьшить переобучение, можно рассмотреть и другие методы регуляризации, такие как L2-регуляризация.

Динамика обучения

- Уменьшение потерь: На протяжении обучения наблюдается стабильно снижающаяся валидационная потеря, что является хорошим признаком улучшения модели. На 40-й эпохе была достигнута самая низкая потеря, что и привело к остановке обучения по причине ранней остановки.
- Ранняя остановка: Этот механизм позволил избежать переобучения, прекратив обучение, когда улучшения стали минимальными. Это помогает сохранить обобщающую способность модели на тестовых данных.

Модель продемонстрировала хорошие результаты, но есть потенциал для улучшения. Текущие результаты показывают, что модель способна распознавать эмоции с высокой точностью, но склонна к предсказанию определённых категорий, что может быть улучшено через дальнейшую настройку и балансировку данных.

Вывод результатов обучения модели показан на рисунке 3.2.

```

Epoch [1/50], Loss: 0.7321, Test Accuracy: 0.7149, Val Loss: 0.7699
Epoch [2/50], Loss: 1.2201, Test Accuracy: 0.7273, Val Loss: 0.7272
Epoch [3/50], Loss: 1.0761, Test Accuracy: 0.7393, Val Loss: 0.7041
Epoch [4/50], Loss: 0.9900, Test Accuracy: 0.7416, Val Loss: 0.6848
Epoch [5/50], Loss: 0.7223, Test Accuracy: 0.7505, Val Loss: 0.6789
Epoch [6/50], Loss: 0.8933, Test Accuracy: 0.7522, Val Loss: 0.6636
Epoch [7/50], Loss: 0.3813, Test Accuracy: 0.7549, Val Loss: 0.6550
Epoch [8/50], Loss: 0.4713, Test Accuracy: 0.7531, Val Loss: 0.6571
Epoch [9/50], Loss: 0.3137, Test Accuracy: 0.7574, Val Loss: 0.6470
Epoch [10/50], Loss: 0.9559, Test Accuracy: 0.7543, Val Loss: 0.6518
Epoch [11/50], Loss: 1.4847, Test Accuracy: 0.7628, Val Loss: 0.6415
Epoch [12/50], Loss: 0.2806, Test Accuracy: 0.7622, Val Loss: 0.6330
Epoch [13/50], Loss: 0.8896, Test Accuracy: 0.7653, Val Loss: 0.6358
Epoch [14/50], Loss: 0.9228, Test Accuracy: 0.7670, Val Loss: 0.6265
Epoch [15/50], Loss: 0.4831, Test Accuracy: 0.7675, Val Loss: 0.6278
Epoch [16/50], Loss: 0.3219, Test Accuracy: 0.7705, Val Loss: 0.6288
Epoch [17/50], Loss: 0.4871, Test Accuracy: 0.7690, Val Loss: 0.6189
Epoch [18/50], Loss: 1.0701, Test Accuracy: 0.7731, Val Loss: 0.6166
Epoch [19/50], Loss: 0.3872, Test Accuracy: 0.7698, Val Loss: 0.6236
Epoch [20/50], Loss: 0.4358, Test Accuracy: 0.7696, Val Loss: 0.6178
Epoch [21/50], Loss: 0.8898, Test Accuracy: 0.7723, Val Loss: 0.6174
Epoch [22/50], Loss: 0.5897, Test Accuracy: 0.7723, Val Loss: 0.6136
Epoch [23/50], Loss: 0.7706, Test Accuracy: 0.7758, Val Loss: 0.6080
Epoch [24/50], Loss: 0.2957, Test Accuracy: 0.7737, Val Loss: 0.6122
Epoch [25/50], Loss: 0.2735, Test Accuracy: 0.7758, Val Loss: 0.6062
Epoch [26/50], Loss: 0.8127, Test Accuracy: 0.7757, Val Loss: 0.6065
Epoch [27/50], Loss: 1.3259, Test Accuracy: 0.7780, Val Loss: 0.6040
Epoch [28/50], Loss: 0.3216, Test Accuracy: 0.7780, Val Loss: 0.6079
Epoch [29/50], Loss: 0.9233, Test Accuracy: 0.7757, Val Loss: 0.6030
Epoch [30/50], Loss: 0.2464, Test Accuracy: 0.7788, Val Loss: 0.6020
Epoch [31/50], Loss: 0.2507, Test Accuracy: 0.7798, Val Loss: 0.6058
Epoch [32/50], Loss: 0.8369, Test Accuracy: 0.7803, Val Loss: 0.6023
Epoch [33/50], Loss: 0.7851, Test Accuracy: 0.7797, Val Loss: 0.6018
Epoch [34/50], Loss: 0.1891, Test Accuracy: 0.7794, Val Loss: 0.6035
Epoch [35/50], Loss: 0.6236, Test Accuracy: 0.7824, Val Loss: 0.5988
Epoch [36/50], Loss: 0.4315, Test Accuracy: 0.7806, Val Loss: 0.5984
Epoch [37/50], Loss: 0.3408, Test Accuracy: 0.7791, Val Loss: 0.6037
Epoch [38/50], Loss: 0.1185, Test Accuracy: 0.7791, Val Loss: 0.5992
Epoch [39/50], Loss: 0.4948, Test Accuracy: 0.7786, Val Loss: 0.5970
Epoch [40/50], Loss: 0.2418, Test Accuracy: 0.7809, Val Loss: 0.5960
Early stopping
Test Accuracy: 0.7824

```

Рисунок 3.2 — Результаты обучения модели

Листинг программы 3

```

import torch
import torch.nn as nn
import torch.optim as optim
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from torch.utils.data import DataLoader, TensorDataset

# Извлечение признаков и меток
X = np.array(data_prepared['features'].tolist())
y = data_prepared['emotion'].values

# Кодирование меток эмоций
label_encoder = LabelEncoder()
y_encoded = label_encoder.fit_transform(y)

# Разделение данных на обучающую и тестовую выборки
X_train, X_test, y_train, y_test = train_test_split(X, y_encoded, test_size=0.2,
random_state=42)

```

```

# Нормализация признаков
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Преобразование данных в тензоры
X_train_tensor = torch.tensor(X_train_scaled, dtype=torch.float32)
y_train_tensor = torch.tensor(y_train, dtype=torch.long)
X_test_tensor = torch.tensor(X_test_scaled, dtype=torch.float32)
y_test_tensor = torch.tensor(y_test, dtype=torch.long)

# Создание DataLoader для PyTorch
train_dataset = TensorDataset(X_train_tensor, y_train_tensor)
test_dataset = TensorDataset(X_test_tensor, y_test_tensor)

train_loader = DataLoader(train_dataset, batch_size=32, shuffle=True)
test_loader = DataLoader(test_dataset, batch_size=32, shuffle=False)

# Определение модели
class ImprovedEmotionClassifier(nn.Module):
    def __init__(self, input_dim, hidden_dim, output_dim):
        super(ImprovedEmotionClassifier, self).__init__()
        self.fc1 = nn.Linear(input_dim, hidden_dim)
        self.dropout1 = nn.Dropout(0.3)
        self.fc2 = nn.Linear(hidden_dim, hidden_dim)
        self.dropout2 = nn.Dropout(0.3)
        self.fc3 = nn.Linear(hidden_dim, hidden_dim // 2)
        self.fc4 = nn.Linear(hidden_dim // 2, output_dim)

    def forward(self, x):
        x = torch.relu(self.fc1(x))
        x = self.dropout1(x)
        x = torch.relu(self.fc2(x))
        x = self.dropout2(x)
        x = torch.relu(self.fc3(x))
        x = self.fc4(x)
        return x

# Параметры модели
input_dim = X_train_scaled.shape[1]
hidden_dim = 256
output_dim = len(label_encoder.classes_)

# Создание модели, определение функции потерь и оптимизатора
model = ImprovedEmotionClassifier(input_dim, hidden_dim, output_dim)
criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(model.parameters(), lr=0.001)

# Планировщик скорости обучения
scheduler = optim.lr_scheduler.ReduceLROnPlateau(optimizer, mode='min', factor=0.5,
patience=3, verbose=True)

# Обучение модели с ранней остановкой
num_epochs = 50

```



```

patience = 5
best_accuracy = 0
patience_counter = 0

for epoch in range(num_epochs):
    model.train()
    for X_batch, y_batch in train_loader:
        optimizer.zero_grad()
        outputs = model(X_batch)
        loss = criterion(outputs, y_batch)
        loss.backward()
        optimizer.step()

    model.eval()
    correct = 0
    total = 0
    val_loss = 0
    with torch.no_grad():
        for X_batch, y_batch in test_loader:
            outputs = model(X_batch)
            val_loss += criterion(outputs, y_batch).item()
            _, predicted = torch.max(outputs.data, 1)
            total += y_batch.size(0)
            correct += (predicted == y_batch).sum().item()

    accuracy = correct / total
    val_loss /= len(test_loader)
    print(f'Epoch [{epoch+1}/{num_epochs}], Loss: {loss.item():.4f}, Test Accuracy: {accuracy:.4f}, Val Loss: {val_loss:.4f}')

    scheduler.step(val_loss)

    if accuracy > best_accuracy:
        best_accuracy = accuracy
        patience_counter = 0
        torch.save(model.state_dict(), 'best_model_improved.pth')
    else:
        patience_counter += 1
        if patience_counter >= patience:
            print("Early stopping")
            break

# Загрузка лучшей модели
model.load_state_dict(torch.load('best_model_improved.pth'))
# Оценка модели
model.eval()
correct = 0
total = 0
with torch.no_grad():
    for X_batch, y_batch in test_loader:
        outputs = model(X_batch)
        _, predicted = torch.max(outputs.data, 1)
        total += y_batch.size(0)
        correct += (predicted == y_batch).sum().item()
accuracy = correct / total
print(f'Test Accuracy: {accuracy:.4f}')

```

4 Выводы о работе

Практическая значимость работы:

1. Распознавание эмоций:

Автоматическое распознавание эмоций имеет широкое применение в различных областях, таких как психология, маркетинг, образование и здравоохранение. Модель, способная точно определять эмоции, может помочь в создании более персонализированных и эмпатичных взаимодействий.

2. Человеко-компьютерное взаимодействие:

Улучшение систем распознавания эмоций может значительно повысить качество взаимодействия между человеком и машиной. Например, голосовые помощники и чат-боты могут стать более чуткими к эмоциональному состоянию пользователя.

3. Мониторинг психического здоровья:

Точные модели распознавания эмоций могут быть использованы для мониторинга и анализа психического состояния человека, что важно для профилактики и лечения различных психологических расстройств.

Научная значимость работы:

1. Исследование моделей нейронных сетей:

Работа предоставляет важные данные о том, как различные архитектуры и гиперпараметры моделей нейронных сетей влияют на их способность к классификации сложных данных. Это может быть полезно для дальнейших исследований в области машинного обучения и искусственного интеллекта.

2. Балансировка данных и регуляризация:

Исследования по улучшению моделей классификации через балансировку данных и применение методов регуляризации могут помочь в разработке более устойчивых и обобщающих моделей.

Техническая значимость работы:

1. Разработка и оптимизация алгоритмов:

Работа демонстрирует процесс разработки и оптимизации алгоритмов классификации, включая выбор архитектуры, методов оптимизации и регуляризации, что важно для создания эффективных и надежных систем машинного обучения.

2. Инженерные подходы:

Подходы, использованные в данной работе, могут быть применены к другим задачам классификации в различных областях, таких как обработка изображений, текстов или звуков.

Социальная значимость работы:

1. Улучшение пользовательского опыта:

Точные системы распознавания эмоций могут значительно улучшить пользовательский опыт в различных приложениях, делая их более отзывчивыми и человечными.

2. Снижение стресса и повышение благополучия:

Применение таких систем в личных устройствах или рабочих местах может помочь людям лучше управлять своим эмоциональным состоянием, что может способствовать снижению стресса и повышению общего благополучия.

Эта работа имеет важное значение на нескольких уровнях: практическом, научном, техническом и социальном. Разработка и улучшение моделей классификации эмоций открывают новые возможности для применения искусственного интеллекта в различных сферах, улучшая

взаимодействие между человеком и технологией, а также способствуя общему благополучию общества

Заключение

В рамках данной курсовой работы была поставлена цель разработки нейронной сети для предсказания настроений на основе аудио признаков.

В процессе работы были решены следующие задачи исследования:

1. Сбор и подготовка данных: Был создан набор данных аудио файлов с разметкой эмоций, а также проведено извлечение релевантных аудио признаков. Это позволило подготовить данные для обучения и тестирования модели.
2. Разработка базовой модели: На основе стандартных архитектур и методов была создана базовая нейронная сеть для классификации эмоций в аудио файлах. Эта модель служила отправной точкой для дальнейших улучшений.
3. Оптимизация и улучшение модели: Были применены различные методы регуляризации, изменения архитектуры сети, гиперпараметрической оптимизации и другие техники с целью повышения точности предсказаний. Этот этап позволил создать более эффективную модель.
4. Оценка и анализ результатов: Проведены эксперименты, оценена производительность модели на тестовых данных и проведен анализ полученных результатов. Точность и стабильность предсказаний были основными метриками оценки.

В результате работы была разработана модель, способная эффективно классифицировать эмоции, выраженные в аудио файлах. Это открывает новые возможности для применения искусственного интеллекта в области анализа и обработки звука, а также для создания более эмоционально чувствительных систем и приложений.

Дальнейшие исследования могут быть направлены на улучшение модели путем использования более сложных архитектур нейронных сетей, расширения набора данных для обучения и тестирования, а также на адаптацию модели для специфических прикладных областей, таких как медицинская диагностика или образование.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Dusha Data Set[Электронный ресурс]: Режим доступа: <https://habr.com/ru/companies/sberdevices/articles/715468/>, свободный (Дата последнего обращения: 05.04.2024)
2. Scikit-learn[Электронный ресурс]: Режим доступа: <https://scikit-learn.org/stable/index.html>, свободный (Дата последнего обращения: 05.06.2024)
3. PyTorch documentation[Электронный ресурс]: Режим доступа: <https://pytorch.org/docs/stable/index.html>, свободный (Дата последнего обращения: 05.05.2024)
4. Librosa documentation[Электронный ресурс]: Режим доступа: <https://librosa.org/doc/latest/index.html>, свободный (Дата последнего обращения: 06.05.2024)
5. librosa.feature.mfcc[Электронный ресурс]: Режим доступа: <https://librosa.org/doc/main/generated/librosa.feature.mfcc.html>, свободный (Дата последнего обращения: 05.05.2024)
6. Классификация аудиофайлов с библиотекой Librosa[Электронный ресурс]: Режим доступа: <https://habr.com/ru/companies/otus/articles/741080/>, свободный (Дата последнего обращения: 05.05.2024)
7. Кузьмин В. И. Методы анализ данных, лекции, РТУ, МИРЭА, 2021 учебный год.
8. Регистрация и авторизация в Django[Электронный ресурс]: Режим доступа: <https://docs.djangoproject.com/en/4.0/topics/auth/default/>, свободный (Дата последнего обращения: 06.04.2022)
9. Кузьмин В.И., Гадзаов А.Ф. Методы построения моделей по эмпирическим данным: Учебное пособие / М.: МИРЭА, 2012. – 94 с.
10. Кузьмин В. И., Гадзаов А.Ф. Модели и методы научно-технического прогнозирования: Учебное пособие /. – М.: МИРЭА, 2016. – 90 с.

11. Кузьмин В. И., Гадзаов А.Ф. Технический анализ: Учебное пособие /. – М.: МИРЭА, МГУПИ, 2015. – 71 с.
12. Анализ аудиоданных (часть 1) [Электронный ресурс]: Режим доступа: <https://habr.com/ru/articles/668518/>, свободный (Дата последнего обращения: 05.05.2024)
13. Анализ аудиоданных (часть 2) [Электронный ресурс]: Режим доступа: <https://habr.com/ru/articles/670676/>, свободный (Дата последнего обращения: 06.05.2024)
14. Анализ аудиоданных (часть 3) [Электронный ресурс]: Режим доступа: <https://habr.com/ru/articles/672094/>, свободный (Дата последнего обращения: 07.05.2024)

