(Слайд 2) В наше время можно увидеть широкое распространение аудиоинформации на различных носителях: это и радио, и сеть Интернет, и компакт-диски, и многое другое. Набирают популярность различные стриминговые платформы, которые дают возможность прослушивания музыкальных каталогов по подписке, например, Spotify, VK combo и Яндекс.Музыка. В связи с этим возникает множество направлений обработки музыки, одним из которых является ее поиск по фрагменту. Самым известным приложением, позволяющим решать эту задачу является “Shazam”, с помощью которого миллионы людей по всему миру могут найти нужную им песню лишь по ее отрывку. Принцип его работы заключается в том, что хеши, извлекаемые из записанного трека, сравниваются с имеющимися хешами из базы данных с последующими уточнениями. Однако, в том же “Shazam” можно столкнуться с проблемой сопоставления оригинальной записи и, например, обычного акапельного исполнения. Отсюда может появиться задача создания аудио-поисковой системы, то есть системы поиска музыки по фрагменту главной мелодии, напетой или сыгранной на каком-либо инструменте. Один из способов решения этого вопроса - распознавание нот. Музыку можно оценивать по-разному. Наиболее техническим способом является гармонический анализ, который помогает понять инструменты и методы, используемые композитором при написании мелодии. Современный мир позволяет автоматизировать этот процесс, к тому же сделать его быстрым и точным. Так, задача распознавания нот может быть решена комбинацией из гармонического анализа и нейронных сетей за счет знания законов музыки и высоких вычислительных мощностей. Достаточно распознать главную мелодию, чтобы, например, имея банк записей нот различных музыкальных инструментов, суметь заменить звучащий инструмент или голос на необходимый.

(Слайд 3) Таким образом, целью исследования будем рассматривать разработку метода распознавания нот мелодии путем извлечения основных гармоник. При этом необходимо свести обработку звука к работе с изображением.

(Слайд 4) Для исследования необходимо выбрать архитектуру нейронной сети. Существует множество различных вариантов, которые могут нам подойти. В большинстве своем самыми используемыми в таких сетях являются сверточные слои и различные вариации слоев подвыборки (пулинга), будь то перекрывающий пулинг, макспулинг или средний пулинг. Сверточные слои осуществляют операцию дискретной свертки для формирования выходной карты признаков для фильтрации маловажных деталей и выделения существенного. Слои пулинга применяются для уменьшения размерности входного тензора, что позволяет сократить количество дальнейших вычислений, а также добавляет устойчивость к небольшим сдвигам объектов на изображении. Для устранения переобучения чаще применяется метод прореживания, т.е. слой Dropout, для исключения на каждой итерации изменения весовых коэффициентов части весов с некоторой заданной вероятностью. На выходе ставится слой softmax, который переводит наборы чисел в вероятности, сумма которых равна 1. Итоговому значению соответствует максимальная вероятность. Формула softmax показана под номером (1).

Для примера рассмотрим сеть LeNet5, схема которой показана на рисунке 1. Она включает 7 слоев, в дополнение к входному слою, 3 из них – сверточные, два – слои подвыборки. На выходе – слой softmax с 10 возможными значениями, соответствующими цифрам от 0 до 9. Ключевое преимущество LeNet5 - возможность сохранять параметры и результаты вычислений, по сравнению с использованием каждого пикселя в качестве отдельных входных данных для большой многослойной сети. Изображения сильно коррелированы пространственно, поэтому в первом слое не используются пиксели, что означает, что использование в качестве входных свойств отдельных пикселей не даст применить преимущества этих корреляций.

(Слайд 5) Мы не будем опираться на различие октав, поэтому основываемся на 12 основных полутонах: A (ля), G# (соль-диез), G (соль), F# (фа-диез), F (фа), E (ми), D# (ре-диез), D (ре), C# (до-диез), C (до), B (си), A# (ля-диез). Таким образом, в нашем распоряжении 12 классов, из которых нужно выбрать один, подходящий для заданной ноты.

В случае бинарной классификации целевая переменная кодировала бы принадлежность к положительному или отрицательному классу. При многоклассовой классификации для линейных моделей осуществляется переход к набору бинарных классификаций. Тем не менее, идея линейной классификации – возможность разделения пространства признаков на 2 полупространства гиперплоскостью и прогнозирования в каждом из них одного из двух значений целевого класса.

По сравнению с классификацией, регрессия предсказывает некоторое непрерывное значение, при этом оно может быть как целым, так и с плавающей запятой. Логистическая регрессия, в отличие от обычной линейной, оценивает, к какой категории данных принадлежит конкретная точка, путем вычисления вероятности принадлежности к классу. Именно прогнозирование вероятности во многих задачах является важным требованием. Поэтому логистическая регрессия – это частный случай линейного классификатора. Для многоклассовой задачи строятся линейные модели (2) по количеству классов, каждая из которых дает оценку принадлежности к конкретному классу. С помощью оператора softmax эти оценки преобразовываются в вероятность путем нормировки вектора оценок. Вероятность к-го класса будет тогда вычисляться по формуле (3).

(Слайд 6) Далее будем придерживаться следующей схемы действий для получения основной гармоники мелодии:

1. Подготовим базу данных для распознавания;
2. На основе базы данных сгенерируем спектрограммы;
3. Сформируем архитектуру нейронной сети для классификации нот;
4. Подадим спектрограммы на вход нейронной сети и обучим модель;
5. Сопоставим полученные номера классов и ноты.

(Слайд 7) Для распознавания нот в произведениях необходимо сначала найти базы данных. В рассматриваемом случае данные представляют из себя аудиофайлы формата WAV, являющиеся моно-сигналами, при этом для каждого файла должна иметься информация о частоте/высоте тона, времени появления и времени окончания/длительности каждой ноты в секундах. Если в базе даны длительности нот, то их можно легко перевести во время окончания формулой (4). Если вместо частоты дана информация о высоте тона ноты, то будем использовать стандарт midi (5). Частота мелодий стандартно рассматривается равной 44,1 кГц, то есть 44100 отсчета в секунду. Значит, для перевода времени в отсчеты воспользуемся формулой (6).

(Слайд 8) В рамках проделанной работы рассматривалось 2 набора данных: «База данных с маркировкой звука флейты для автоматической транскрипции музыки» и «CSD: набор данных детских песен для исследования певческого голоса», при этом вторая база также разделялась на два подкаталога английских и корейских песен. Обработка этих наборов состоит в следующем:

1. Сначала загружаем данные и метки;
2. Переразбиваем данные на маленькие интервалы, пригодные для обучения;
3. После переразбиваем метки согласно данным.

(Слайд 9) При разбиении аудиофайлов на интервалы время переведем в отсчеты. Получившиеся фрагменты необходимо распределить на каталоги, соответствующие конкретным нотам. Такое распределение осуществимо из соответствия нот и частот. Математическая формула частоты для всего звукоряда имеет вид (7). Для нахождения количества полутонов, на которое заданная частота отстает от частоты камертона, используется формула (8). Для сведения всех нот к 12 основным, необходимо найти значение по формуле (9). При этом, если , то разворачиваем ноту так, чтобы она оказалась выше ноты A. Это отображается во второй строке системы. Нота, соответствующая заданной частоте без учета октав, определяется из значения z по таблице 1.

( Слайд 10) Следующий пункт схемы получения основной гармоники – генерация спектрограмм на основе полученной базы данных. Именно спектрограмма будет являться тем изображением, которое мы подадим на вход нейронной сети.

Спектрограмма — изображение, показывающее зависимость спектральной плотности мощности сигнала от времени. Они применяются для идентификации речи, анализа звуков животных, в различных областях музыки, радио- и гидролокации, обработке речи, сейсмологии и в других областях. Спектрограмма представляет собой матрицу , где соответствует определенной частоте, а каждый из столбцов – спектр фрагмента аудиозаписи. Неформально говоря, элементы спектрограммы характеризуют интенсивность звучания конкретных частот в пределах промежутка времени. Как видно по рисунку 2, наиболее распространенным представлением спектрограммы является двумерная диаграмма: на горизонтальной оси представлено время, по вертикальной оси — частота; третье измерение с указанием амплитуды на определенной частоте в конкретный момент времени представлено интенсивностью или цветом каждой точки изображения.

Одним из способов получения спектрограммы является вычисление оконного преобразования Фурье, представленного формулой (10). Аудиосигнал постоянно меняется, поэтому для упрощения мы предполагаем, что на коротких временных масштабах аудиосигнал статически не сильно меняется. Оконное преобразование Фурье также называют коротким (или кратковременным) преобразованием Фурье (STFT). В качестве весового окна можно использовать различные варианты, но наибольшей популярностью пользуется окно Хэмминга (11).

(Слайд 11) Вместо спектра самого сигнала можно использовать спектр его мощности, что мотивировано человеческой ушной улиткой, которая вибрирует в разных местах в зависимости от частоты входящих звуков. В зависимости от расположения в вибрирующей улитке срабатывают различные нервы, сообщая мозгу, что присутствуют определенные частоты. Оценка спектрограммы выполняет аналогичную работу для нас, определяя, какие частоты присутствуют в кадре.

Спектральная оценка спектрограммы содержит много информации, не требуемой для автоматического распознавания нот. В частности, улитка не может различить разницу между двумя близко расположенными частотами. Этот эффект становится более выраженным по мере увеличения частот. Поэтому мы будем брать скопления блоков спектрограммы и просуммируем их, чтобы получить представление о том, сколько энергии существует в различных частотных областях. Это выполняется с помощью банков фильтров Мела: по мере увеличения частоты фильтры расширяются, поскольку мы становимся менее обеспокоенными вариациями. Нас интересует только приблизительно, сколько энергии приходится на каждый фрагмент. Шкала мела направлена на то, чтобы имитировать нелинейное восприятие звука человеческим ухом, будучи более различимой на более низких частотах и менее различимой на более высоких частотах. Конвертировать Герцы и Мелы между собой можно с помощью формул (12) и (13). Мы будем использовать банк треугольных фильтров: каждый фильтр имеет треугольную форму с откликом 1 на центральной части и линейно уменьшается к 0, пока не достигнет центральных частот двух соседних фильтров, где отклик равен 0. Моделирование таких фильтров можно осуществить с помощью уравнения (14). На рисунке 3 показан пример банка треугольных фильтров. Как только у нас есть энергии фильтрующего банка, мы берем логарифм их, получая кепстр или мел-кепстральные коэффициенты MFCC — представление сигнала, грубо говоря, в виде особого спектра, из которого с помощью различных фильтраций и преобразований удалены незначительные для человеческого слуха компоненты.

(Слайд 12) В качестве архитектуры сети рассмотрим предложенную Филипом Корзениовски в диссертации «Harmonic Analysis of Musical Audio using Deep Neural Networks». Такая сеть состоит из 3 скрытых слоев из 512 узлов с функцией активации ReLU и выходного слоя из 12 узлов по количеству полутонов. Параметры нейронной сети показаны на слайде. Модель сети представлена на рисунке 4. Как из нее видно, в качестве входных данных используются спектрограммы.

(Слайд 13) Для получения спектрограмм используем 2 подхода: будем сначала использовать спектр сигнала, а потом мел-кепстр. Назовем эти модели моделью 1 и моделью 2, соответственно. Данные будем подавать пакетами по 64 вектора.

Человеческое ухо не слышит колебания меньше 20 Гц, поэтому эта граница будет задавать максимальный размер окна. Для частоты дискретизации 44100 отсчетов/сек. получаем размер окна 2205. Для красоты возьмем 2048 отсчетов, что составляет примерно 21.5 Гц – это и будет первоначальным размером фрейма для оконного преобразования Фурье. Шаг смещения фрейма при этом возьмем равным половине размера кадра. После оконного преобразования Фурье получаем множество коротких спектров, длина которого вычисляется по формуле (16). Множество таких спектров составит одну спектрограмму. На вход нейросети подается несколько соседних спектрограмм. Будем брать, как предлагал Корзениовски, 15 последовательных спектрограмм. Если размер файла не позволяет это осуществить, то будем дополнять спектрограмму последним столбцом до нужного количества. Таким образом, размерность входа для нейросетевой модели – 1025х15.

При получении результатов, номера классов будем сопоставлять с нотой в соответствии с таблицей 1.

(Слайд 14) Вся дальнейшая работа производилась с помощью языка программирования Python. Сначала рассмотрим модель 1. На рисунке 5 (а) представлена спектрограмма, соответствующая некоторой аудиозаписи ноты ре-диез. По оси абсцисс отложены номера отсчетов, по оси ординат – номера спектрограмм. На тестовой выборке модель с такими входными данными, обученная на 100 эпохах, дает примерно 95% точности. Однако, если посмотреть на графики штрафа и точности для тренировочной и проверочной выборок (рисунки 5 (б) и 5 (в), соответственно), то можно увидеть переобучение. Для проверки работоспособности модели также была совершена попытка предсказать ноту «ми», которая была успешно пройдена, что показывает гистограмма 5 (г). Исходя из матрицы ошибок (рисунок 5 (д)), можно сказать, что в основном модель работает правильно. Время работы модели составило 12 минут 45 секунд.

(Слайд 15) Далее рассмотрим результаты модели 2. На рисунке 6 (а) представлена спектрограмма, соответствующая некоторой аудиозаписи ноты ре-диез. На этот раз по оси абсцисс отложены номера спектрограмм, в то время как по оси ординат показаны номера треугольных фильтров. Точность на тестовой выборке при обучении на 100 эпохах чуть меньше 92%, что ниже результата модели 1. По графикам штрафа и точности для тренировочной и проверочной выборок (рисунки 6 (б) и 6 (в), соответственно) видно, что и при таком подходе присутствует переобучение. Тем не менее, оно начинается позже, чем у модели 1. При проверке распознавания ноты «ми» из гистограммы 6 (г) можно заметить, что есть небольшая вероятность спутать эту ноту с ее соседней – нотой «фа». А матрица ошибок 6 (д) показывает, что модель, использующая второй подход получения спектрограммы, дает больше неверных ответов, чем в предыдущем случае. Однако, время работы модели составило 3 минуты 25 секунд, что почти в 4 раза меньше, чем для модели с первым подходом.

(Слайд 16) Также рассмотрим результаты для демонстрационного примера, который представляет из себя моно-запись короткой гитарной мелодии длительностью 29 секунд. Графики сравнения реальных нот и нот, полученных в результате распознавания моделями 1 и 2 приведены на слайде. Для распознавания аудио-сигнал разделялся на равные по времени фрагменты, количество которых равно количеству нот в мелодии (в оригинальной записи расстояния между нотами не равны). Как можно заметить, модель 1 дает более точные результаты, хотя сама точность обоих моделей довольно низкая.

(Слайд 18) Таким образом, можно на данный момент сделать 2 промежуточных вывода: оба подхода дают неплохую точность распознавания нот при обучении, но при демонстрации их точность низкая, и при использовании обоих подходов присутствует переобучение. В дальнейших исследованиях будут проводиться следующие работы в отношении модели 2:

1. Изменение шага окна;
2. Изменение количества кадров, подаваемых на вход нейросети;
3. Борьба с переобучением;
4. Поиск оптимальных треугольных фильтров;
5. Использование ДКП (дискретного косинусного преобразования) вместо STFT.