Проект на тему: «Определение музыкального темпа»

В нашем проекте мы реализуем продукт, позволяющий определять музыкальный темп с использованием методов глубокого обучения.   
Данная тема показалась нам интересной и актуальной, так как в повседневной жизни люди постоянно сталкиваются с продуктами использующими подобные методы при прослушивании или поиске музыки(например, Shazam, различные стриминговые сервисы и т.д.).  
 Основная цель проекта - создание приложения, позволяющего определять музыкальный темп(BPM).  
 Задачи, которые стояли в работе:  
1. Изучить статьи, связанные с работой с аудиофайлами в целом и определением темпа в частности;  
2. Определить методы технической реализации модели;  
3. Построить нейронную сеть, решающую основную задачу;  
4. Создать сайт, позволяющий пользователю работать с моделью.  
 Для реализации первой задачи, мы ознакомились с несколькими статьями. Мы изучили общие методы, используемые в работах, но для подробного изучения остановились на статье:   
"A SINGLE-STEP APPROACH TO MUSICAL TEMPO ESTIMATION USING  
A CONVOLUTIONAL NEURAL NETWORK" -  
Hendrik Schreiber, Meinard Muller, 2018.  
 Изучение работы сместило нас в сторону выбора классификации, а не регрессии как основной задачи. Авторы в своей работе реализуют одношаговый метод оценки темпа с использованием CNN. Входные данные представляются в виде Mel-спектограмм. Также в работе реализуется интересная аугментация в виде масштабирования по времени с корректирующими методами и случайной обрезкой. Из статьи мы подчеркнули идею использования в виде метрики 3 различных accuracy. Первый представляет собой точное предсказание величины BPM(Accurac0), второй отражает процент треков, где предсказанный BPM отличается от эталонного не более чем на 4%(Accuracy1), третий совпадает со вторым и дополнительно учитывает гармонические ошибки - ситуация когда предсказанный темп отличается от истинного в кратное число раз(Accuracy2). В работе использовался несколько отличный от нашего набор данных - мы старались сделать их как можно разнообразнее (представить как больше различных жанров) и поэтому мы работали с ними немного иначе. В заключение, одношаговый подход CNN реализуемый в модели нацелен на простоту и эффективность, поэтому подходит для использования в учебном проекте.  
  
 Сделаем небольшое теоретическое углубление. В основе работы лежит спектральное представление звука, а конкретно Mel-спектограммы. В общем спектральное представление - это способ представления аудиосигнала в частотной области, который позволяет анализировать его частотные компоненты меняющиеся во времени. Для его получения используется преобразование Фурье, представляющее аудио как спектр, где по оси Х лежит частота, а по Y - амплитуда. Мы же работали с Mel-спектрограммами - разновидность спектрограммы, в которой частотная ось на основе исследований(человеческое ухо воспринимает частоты нелинейно: различия между низкими частотами ощущаются сильнее, чем между высокими) преобразована в Mel-шкалу, приближенную к человеческому восприятию звука. Использование именно такого представления эффективно, так как они компактно представляют ритмическую и частотную информацию, сохраняя ключевые особенности связанные с темпом. Кроме того, они устойчивы к вариациям в высоте тона, что облегчает процесс опредения BPM.   
   
 Следующей задачей стал отбор данных для обучения модели. Изначально мы решили использовать исключительно данные с указанными метками BPM, что усложнило процесс поиска: большинство наборов либо вовсе не содержат таких данных, либо содержат их в усечённом виде. Мы нашли и обработали 3 датасета - Groove(набор содержит записи ударных партий, включая при этом различные музыкальные стили), FMA(его пришлось обрезать до 5000 треков из-за отсутствия меток BPM; сами данные включают различные жанры, исполнителей, альбомы), GTZAN(классический датасет из 1000 треков в 10 жанрах; датасет является самым удобным в использовании, но не велик размером). Уже после первичного обучения модели мы решили провести эксперимент и самостоятельно разметить дополнительные данные с помощью python-библиотеки librosa.

Для этого был выбран датасет Ballroom, представляющий собой аудио для бальных танцев с чёткими ритмическими структурами. После определения меток BPM и дообучения модели на новых данных метрика выросла(с 0.15 до 0.22 для Accuracy0, с 0.45 до 0.55 для Accuracy1), поэтому мы посчитали эксперимент удачным и оставили данные для обучения.   
  
 Следующим этапом стало непосредственно написание кода нейронной модели.

Начнем описание кода с аугментаций. Чтобы обучение проходило в максимально приближенном к реальности мы использовали следующие аугментации : во-первых мы добавили частотную аугментацию (frequency\_augmentation). Этот метод применяет случайные корректировки усиления к отдельным частотным полосам Mel-спектрограммы. Кроме того мы добавили шум, который имитирует реальные условия записи, где обычно присутствует фоновый шум различной интенсивности - от почти незаметного до существенного . Следующая аугментация это случайное усиление (random\_gain).Метод применяет к сигналу коэффициент усиления, случайно выбираемый из диапазона 0.5-1.5 (то есть от уменьшения громкости в 2 раза до увеличения в 1.5 раза). Кроме того мы добавили сдвиг высоты тона (pitch\_shift), который позволяет модели научиться определять BPM независимо от тональности трека - например, распознавать одинаковый ритм в оригинальной и неоригинальной версиях композиции.

Для извлечения Mel-спектограммы создали функцию (extract\_mel\_spectrogram).Она у нас выполняет основную предобработку аудиофайлов. Если более подробно, то сначала она загружает аудиофайл с помощью librosa.load, затем проверяет длину аудио. Если аудио слишком короткое , то дополняет нулями.В конечно итоге мы на выходе получаем Mel-спектограмму и коэффициент масштабирования.

Далее в нашем коде происходит обработка окон спектограммы с помощью crop\_spectrogram.Она извлекает окно фиксированной длины из спектограммы. Ее полезность в том, что она позволяет уменьшить перестройку модели за счет воздействия на различные части аудио во время обучения.

Также важная часть -это подготовка данных -функция prepare\_data.Она обрабатывает датасеты для обучения, валидации и теста. Для каждого аудио она извлекает Mel-спектограмму с помощью extract\_mel\_spectrogram, затем корректирует BPM с учетом масштабирования времени при условии что аугментация включена, и в конце пропускает файлы, если BPM выходит за пределы диапазона (30–285).

Перейдем к описанию функции mf\_mod\_block.Она применяет свертки , что позволяет захватывать ритмические паттерны на разных временных масштабах(например , быстрые удары и более медленные ритмы)

В нашей модели входной слой у нас принимает Mel-спектограммы. Первичные слои извлекают локальные особенности из спектограммы, такие как ритмические удары или изменения интенсивности . BatchNormalization нормализирует батчи и стабилизирует обучение. Потом применяем четыре раза mf\_mod\_block для извлечения сложных признаков. В следующем блоке используем несоклько слоев для разных целей: Flatten для преобразования в одномерный вектор, Dropout для регуляризации и 2 полносвязных слоя Dense для обработки извлеченных признаков и выполнения классификации. На выходе получаем полносвязный слой с 256 нейронами и активацией softmax для классификации. В качеств оптимизатора используем Adam (learning\_rate=0.001). В коде используем раннюю остановку и снижение скорости обучения, чтобы улучшить производительность модели.

На этапе тестирования модель предсказывает BPM для каждого окна спектограммы. Эти предсказания усредняются для получения глобального темпа в аудио.

Заключительным этапом стала реализация сайта, в который включается построенная модель. Мы изначально решили не просто написать код, но и попробовать представить его в виде конечного продукта, поскольку всегда приятнее видеть свой труд в оформленном виде. Базовое представление о форме будущего сайта включало в себя возможность загрузить туда аудио-дорожку и получить результат по BPM. Для технической реализации использовался фреймворк Flask и с помощью HTML написан простенький сайт. Продукт можно запускать локально на своём устройстве к примеру с помощью командной строки. Итоговая версия сайта "Определитель BPM 2000) выводит на экран усреднённый темп по треку, а также его изменение в разный момент песни. Недоработками, которые хотелось бы устранить, является невозможность проигрывания трека прямо на сайте и достаточно медленная скорость загрузки, которую хотелось бы оптимизировать.  
  
К недостаткам самой модели можно отнести не лучшую отработку на уровнях меньше 80 BPM и больше 180 BPM - в этих областях возрастает как частота гармонических ошибок, так и полностью неправильного опредения тона. Мы предполагаем, что эту проблему можно начать решать с помощью расширения датасета для обучения, особенно за счёт аудио с низкими и высоким BPM.   
  
В процессе работы над проектом мы научились работать с достаточно непривычным форматом данных - аудио. Была разработана и протестирована модель на основе нейронных сетей для оценки темпа музыкальных композиций (BPM) с использованием Mel-спектрограмм в качестве входных данных. Для дальнейшего улучшения планируется расширять датасеты, используемые для обучения и попробовать улучшить архитектуру. Проект демонстрирует перспективность использования методов глубокого обучения в сфере музыкальной сфере для создания рекомендательных систем.

Список литературы:

1. «A SINGLE-STEP APPROACH TO MUSICAL TEMPO ESTIMATION USING  
A CONVOLUTIONAL NEURAL NETWORK» -  
Hendrik Schreiber, Meinard Muller, 2018.  
2. «Music Tempo Estimation via Neural Networks – A   
Comparative Analysis» -  
Mila Soares de Oliveira de Souza, Pedro Nuno de Souza Moura, Jean-Pierre Briot  
3. «Musical Tempo and Key Estimation using Convolutional Neural Networks with Directional Filters» -   
Hendrik Schreiber, Meinard Muller