Отчет по тестовому задания

Шевченко Олег

19 сентября 2021 г.

Аннотация

Реализация модели Voice Activity Detection (VAD).

1 Краткий обзор уже существующих методов

В более классических методах в основнов использовались ручные акустические признаки с моделью GMM или HMM. В современных моделях все больше преобладают глубокие нейронные сети на основе CNN и в большей степени LSTM, которая показывает на данной задаче хорошие результаты https://research.google/pubs/pub46246/. Большинство современных подходов используют Noise-independent training, когда модель обучается с различными шумами и различным уровнем SNR.

В работе были рассмотрены и разробны основные ручные признаки, которые можно использовать для распознавания Vad и не только.

Есть статьи в которых рассматривается более стандартный подход. Так в статье в качестве признаков выступает спектрограмма, а архитектура сети представляет собой несколько RNN слоев с линейным выходным слоем.

Довольно часто пытаются построить систему, которые могла бы работать в реал-тайме и которая рассматривает только настоящий и предыдущие фреймы. Так есть пример данной статьи.

Также встречаются и довольно необычные подходы с использованием boosted DNN и выделением нестандартных входных признаков (Multi-resolution cochleagram (MRCG)) http://web.cse.ohio-state.edu/~wang.77/papers/Zhang-Wang.taslp16_1.pdf

Появляются также подходы и решения с использованием информации о контретных спикерах https://arxiv.org/pdf/1908.04284.pdf. А так же подходы на основе исходных waveform https://arxiv.org/pdf/2006.11139.pdf.

2 Описание выбранной архитектуры и выбранной метрики качества

2.1 Описания алгоритма обучения

Как мы обсуждали теоретическое решение на первом собеседовании, такой же принцип был реализован и здесь. Был взят неразмеченный датасет чистой речи, а так же слабый бейзлайн алгоритм в качестве разметчика. В качество бейзлайн алгоритмы был взят webrtcvad. В webrtc есть 4 уровня, было замечено, что на 0ом уровне этот алгоритм почти все распознает как речь, а на уровне 4 довольно часто предсказывал одни нули (т.е. что в аудио файле вообще не было речи). Поэтому был выбран 1 уровень.

С помощью бейзлайн модели размечались чистые данные, делились на train и test. Также было замечено, что кастомный датасет for_devs имел много шумов, поэтому необходимо было построить робастную систему, устойчивую к шуму. Для этого надо добавить шумов к нашим чистым данным и таким образом сделать заранее размеченный датасет более шумным. В качестве шумов был взят отдельный датасет шумов pnl (подробнее о нем в разделе описание данных).

Так же на собеседовании обсуждалось способ добавление шумов с заданной громкостью в два раза меньшей, чем исходная речь. Signal-2-noise ratio. Это тоже было учтено при добавлении шумов. таким образом, если мы хотим сделать шум в два раза тише исходного сигнала, то snr

будет равен 3. При необходимости могу предоставить расчеты уравнений, из которых я делал выводы. В частности этот процесс довольно подробно описан в этом посте.

Шумы для исходного набора данных добавлялись случайным образом, при этом для тестовых данных они были добавлены один раз и после этого тестовый набор больше не менялся. Для обучения же шумы и параметры их добавления не фиксировались и добавлялись случайным образом, чтобы улучшить обобщающую способность алгоритма.

2.2 Описание данных

В качестве исходного датасета был взят датасет librispeech, который был предложен в описании тестового задания. Из него были взяты только train-100 и train-360. Модуль train-500 не рассматривался так как обладает более шумными записями и потому что занимает намного больше места.

В качестве шумов был взять датасет PNL 100 - это не языковой датасет с различным набором звуков. В нем присутствуют такие шумы и звуки, как улица, сирена, толпа и тд., всего в сумме 100 звуков. Как мне показалось, этот датасет отлично подходит для данной задачи добавления шумов в исходные данные.

2.3 Описание признаков

В качестве признаков были взяты мел-кепстральные коэффициенты mfcc с окном в 25 мс и 10 мс перекрытие, так же как мы обсуждали на собеседовании. Также была добавлена нормализация тензора. Согласно статье CMVN in the model domain добавление нормализации CMVN (Cepstral mean and variance normalization) к mfcc улучшает обобщающую способность систему и повышает робастость, поэтому так же после выделения признаков была добавлены CMVN нормализация.

2.4 Описание модели

Были протестированы несколько моделей: модель только с линейными слоями, LSTM/GRU с несколькими линейными слоями на выходе. Наилучший результат показала архитектура на основе LSTM с 2мя внутренними слоями и 2мя линейными слоями. После обучения модель также была преобразована в формат ONNX для ускорения инференса.

2.5 Метрики качества

В качестве основной метрики качества были выбраны precision и recall. Хотя на собеседовании обсуждались альтернативы, но там шла речь в контексте обсуждения и объяснения метрик менеджеру, в нашем же случае мы можем показывать качество и на precision, recall. Но это не значит, что альтернативные метрики не были рассмотрены вовсе, они так же были рассчитаны и включены в рассмотрение. А именно был показан roc-auc, а также False acceptance ratio (FAR) и False rejection ratio (FRR). По результатам экспериментов на валидации были выбраны следующие пороги для метрик FA и FR: FA=1% 0.95, FR=1% 0.64, FR=FA 0.71. Подробнее на метрики можно посмотреть в ноутбуке. Из данных метрик можно сделать вывод, что представленная модель имеет очень низкий процент неверных отбрасываний.

2.6 Предсказание на тестовом датасете

Предсказания на тестовых данных, а также на сравнение с бейзлайном тоже можно посмотреть в ноутбуке.

2.7 Ссылки на демо и инструкции

1. Инструкции по запуску и обучению
2. Demo-colab

2.8 Возможные улучшения

В процессе создания данной системы были замечены некоторые особенности и возможные улучшения.

Во-первых, webrtc довольно неустойчивая модель и даже на чистых данных иногда невозможно предсказать что он выдаст, возможно в качестве изначального разметчика чистых данных можно взять другие более сильные модели, например, pytorch vad, silero-vad и др.

Во-вторых, добавление новых шумов и аугментаций, более близких к рассматриваемой области, а также импульсные характеристики, которые обсуждали на собеседовании и которые нельзя было использовать в этом тестовом задании.

В-третьих, добавление дополнительных характеристик, т.к. энергию, спектральные центроиды, энергетические характеристики банка фильтров и другие.

В-четвертых, построение более глубокой нейронной сети хотя в этом случае может пострадать скорость распознавания.

В-пятых, добавление более качественных и заранее размеченных данных. Например, TIMIT, https://iqtlabs.github.io/voices/ и https://github.com/gabrielmittag/NISQA/wiki/NISQA-Corpus

3 Список всех источников

1. https://www.recogtech.com/en/knowledge-base/security-level-versus-user-convenience
2. https://github.com/wiseman/py-webrtcvad
3. https://colab.research.google.com/drive/15epQxGh5Pors-qety9KacqisAvfLuVQx#scrollTo=dNvgwWlx3S8U
4. https://www.isca-speech.org/archive_open/archive_papers/robust2004/rob4_38.pdf
5. https://arxiv.org/pdf/1611.09405.pdf
6. https://en.wikipedia.org/wiki/Signal-to-noise_ratio
7. https://medium.com/analytics-vidhya/adding-noise-to-audio-clips-5d8cee24ccb8
8. https://github.com/sleekEagle/audio_processing
9. https://www.codespeedy.com/calculate-signal-to-noise-ratio-in-python/
10. https://www.microsoft.com/en-us/research/wp-content/uploads/2017/04/Tashev-Mirsamadi_DNN-based-Causal-VAD.pdf
11. http://web.cse.ohio-state.edu/pnl/corpus/HuNonspeech/HuCorpus.html
12. https://iopscience.iop.org/article/10.1088/1757-899X/231/1/012042/pdf
13. https://habr.com/ru/post/140828/
14. http://web.cse.ohio-state.edu/~wang.77/papers/Zhang-Wang.taslp16_1.pdf
15. https://arxiv.org/pdf/2006.11139.pdf
16. http://web.cse.ohio-state.edu/~wang.77/papers/CWW.taslp14.pdf
17. https://www.assemblyai.com/blog/end-to-end-speech-recognition-pytorch
18. https://stats.stackexchange.com/questions/272962/are-far-and-frr-the-same-as-fpr-and-fnr-respec
19. https://medium.com/@mustafaazzurri/face-recognition-system-and-calculating-frr-far-and-eer-for-
20. https://arxiv.org/pdf/1908.04284.pdf
21. https://github.com/filippogiruzzi/voice_activity_detection