Отчет по SincNet

Для решения тестового задания мной была выбрана статья "Speaker Recognition From Raw Waveform With SincNet" (https://arxiv.org/pdf/1808.00158.pdf). Весь код связанный с реализацией метода есть в моем гитхабе.

Alex Shemchuk

Все эксперименты можно увидеть по этой ссылке https://wandb.ai/sheminy32/sincnet-exp?workspace=user-sheminy32

Задача

Были проведены эксперименты с SincNet для задачи Speaker Identification.

Описание подхода и отличие от других

SincNet -- это CNN архитектура, в которой заменили первый слой на свертку с sinc фильтрами. Таким образом, SincNet на первом уровне выучивает фильтры с физическим смыслом базируясь просто на waveform. В отличии от CNN, SincNet имеет более быструю сходимость, меньше параметров и менее шумные фильтры (картинка взята из оригинальной статьи):

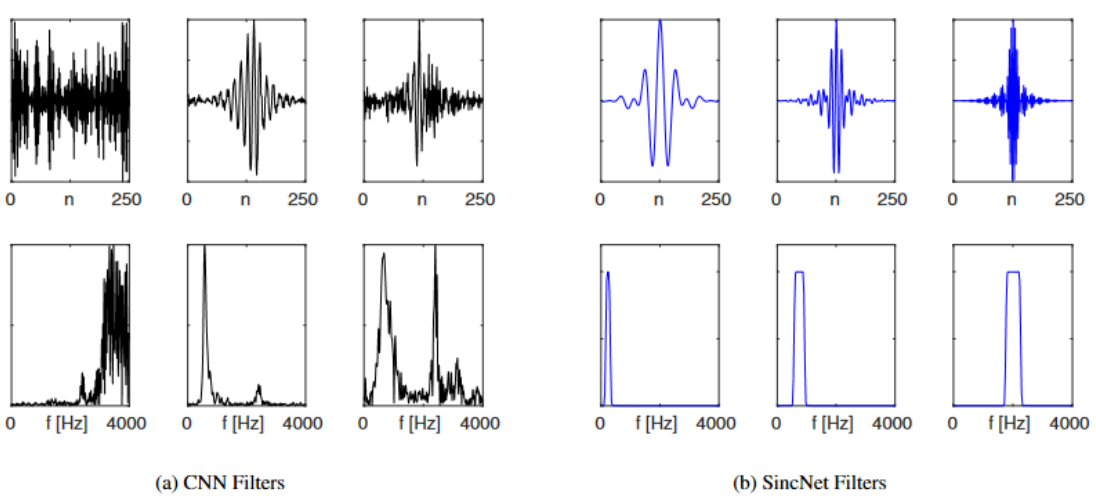


Fig. 2: Examples of filters learned by a standard CNN and by the proposed SincNet (using the Librispeech corpus). The first row reports the filters in the time domain, while the second one shows their magnitude frequency response.

Так же, если сравнивать с подходами основывающимися на handcrafted-features (спектрограмма, FBANK, MFCC), SincNet, как и CNN на waveforms, позволяет избавиться от тюнинга параметров на

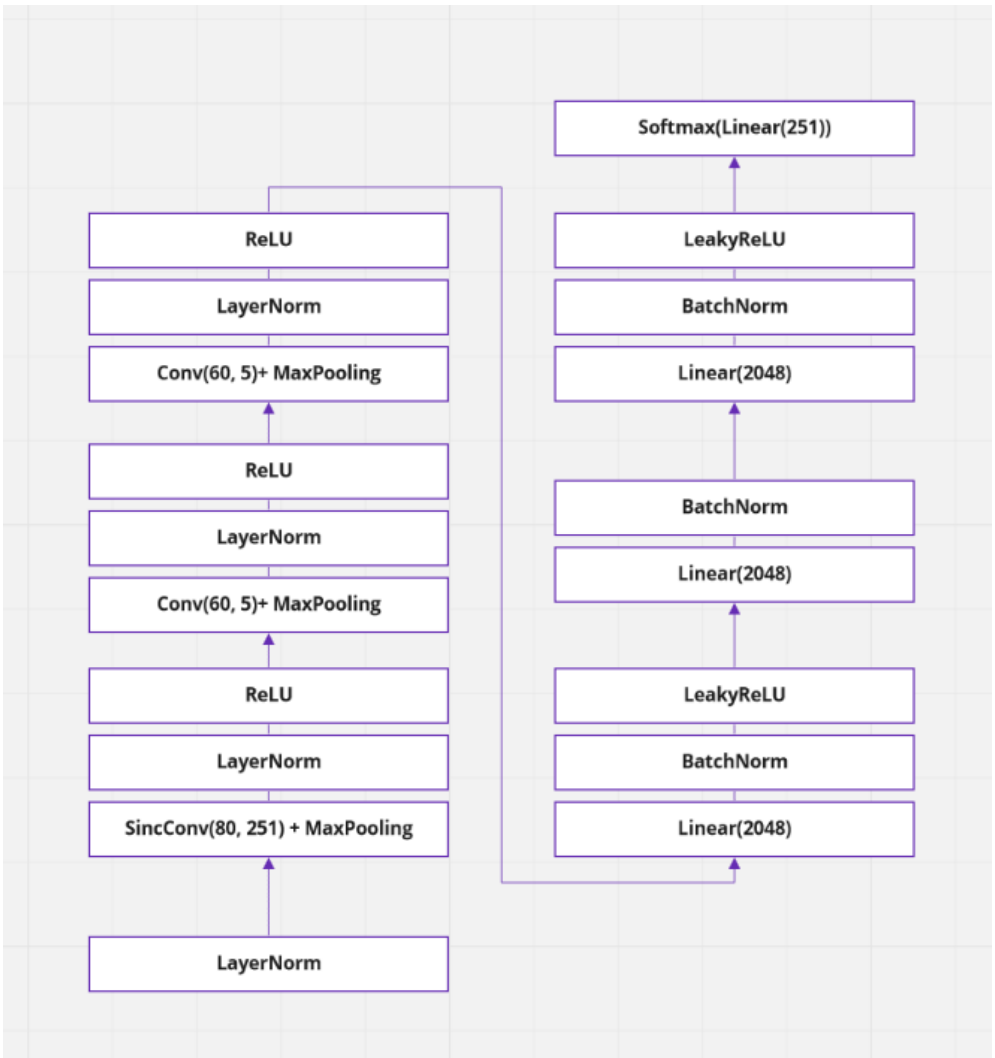
уровне процессинга.

Experimental setup

Для оценки подхода была использована часть датасета LibriSpeech train-clean-100. Для тренировочной выборки были использованы аудио с продолжительностью 12-15с, а для тестовой выборки подбирались аудио с продолжительностью 2-6с. Модель обучалась на чанках этих аудиозаписей по 200мс и оверлэпом 10мс. В отличии от оригинальной статьи, был немного изменен подход к обучению: в коде к оригинальной статье выбиралось 100 батчей по 128 элементов случайным образом, когда как в моей реализации использовалась вся выборка на каждой эпохе, из-за чего за место 1500 эпох, модель обучалась всего 10.

Архитектура

Архитектура SincNet от CNN-Raw отличается лишь первым сверточным слоем. Первым слоем выступает LayerNorm, далее идет backbone из трех блоков. Первая свертка имеет 80 фильтров и kernel_size 251, остальные 60 фильтров и kernel_size 5. Далее идет MLP из 3 блоков. В каждом есть линейный слой с 2048 нейронами, батч норм и LeakyReLU (кроме второго блока).



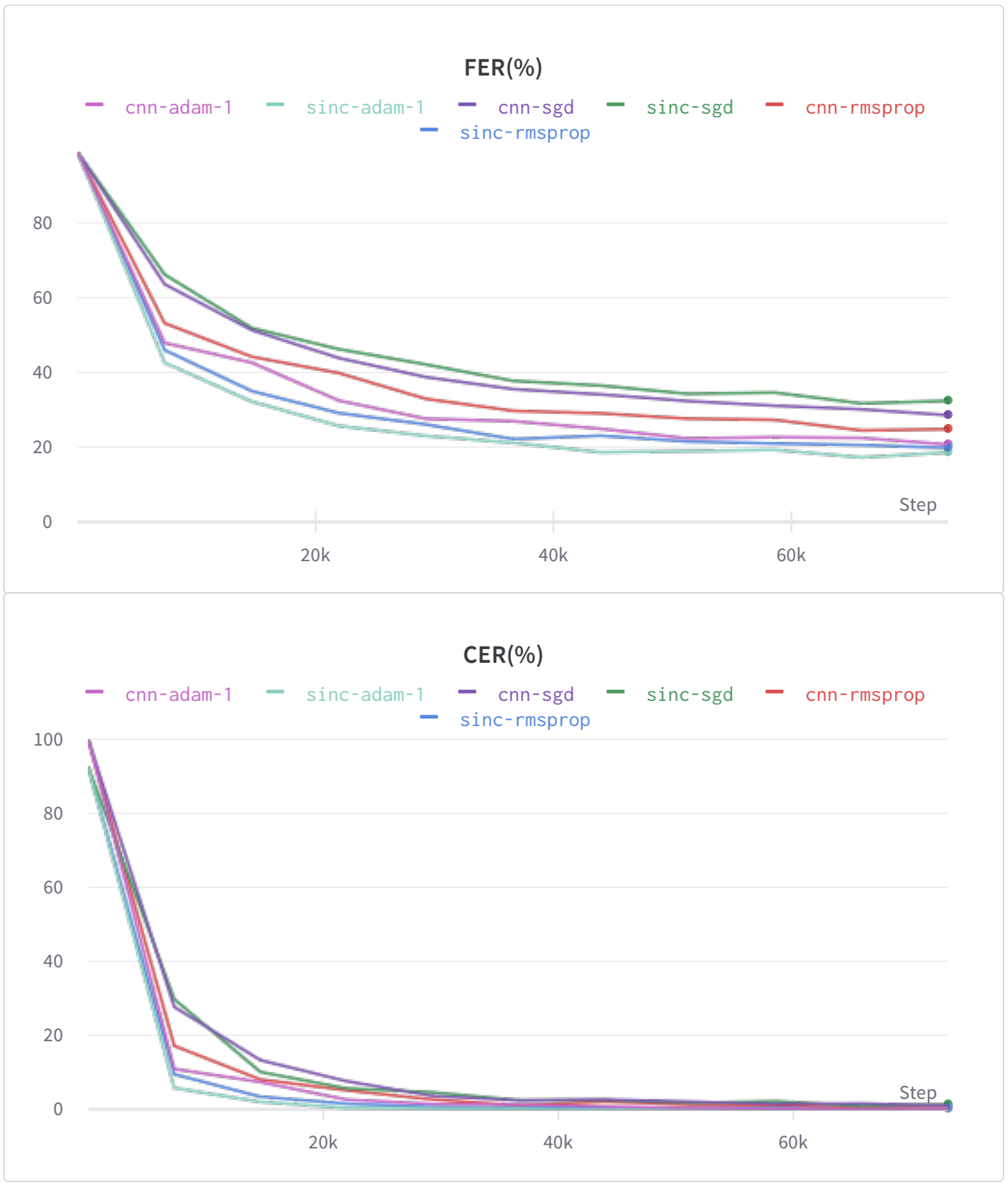
SincNet

Были проведены эксперименты, как с RMSProp предложенным в

статье, так и с Adam и SGD (learning rate для всех 1e-3). Для тренировочной выборки был выбран размер батча 128, а для тестовой -- 256.

Результаты

Во всех экспериментах оценивались две метрики: Frame Error Rate (FER) и Classification Error Rate (CER), где FER -- это ошибка классификации на уровне чанка, а CER -- это ошибка классификации на уровне всей аудиозаписи. Ниже представлены метрики разных сетапов на тестовой выборке для каждой эпохи.



На этом датасете получилось достигнуть наилучших результатов при помощи SincNet с Adam добившись FER 17.46%, когда как для CNN удалось получить FER 20.87%. Так же можно заметить, что SincNet

сошелся на 9 эпохе, когда как CNN на 10.

	Adam	SGD	RMSProp
SincNet	**0.118**	**1.023**	**0.276**
CNN-Raw	0.236	1.22	0.433

CER(%), Результаты для датасета LibriSpeech.

Заметим, что результаты получились лучше, чем в статье, однако модель обучалась на не полном датасете, что могло внести свои корректировки, однако общее положение относительно CNN-Raw осталось такое же.

Заключение и будущее развитие

Эта архитектура универсальная для большого количества задач, поэтому она может быть опробована на различных Time-series задачах. Так же можно попробовать сам SincConv, в других задачах обработки звука. Архитектура этой модели достаточно проста и поэтому можно провести множество экспериментов с архитектурой самой модели (например взять какие-то архитектуры из CV и адаптировать под Sound Processing с использованием SincConv). Так же можно было бы попробовать ArcFace/CosFace/SphereFace на аутпуте для задачи Speaker Identification.

Часть 2*

За последний год мое внимание привлекла статья "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale", где авторы обучают transformer модель для классификации картинок. Из-за своей архитектуры модель учится оценивать какой-то кусок изображения в контексте окружения, что может быть очень полезно для решения задач OCR, где бы я и хотел опробовать подобную архитектуру. Путей улучшения подобного подхода достаточно много, но я бы начал изучать вопрос представления двумерных кусков изображения в одномерном векторе, так как в оригинальной версии для патчей просто делается решейп в вектор. В этом случае можно было бы

попробовать различные свертки. Так же стоит поэкспериментировать с размерами патчей. Пока совершенно не было времени на эксперименты, но в ближайшем будущем планирую заняться :)

Created with ❤️ on Weights & Biases.

https://wandb.ai/sheminy32/sincnet-exp/reports/-SincNet--Vmlldzo3OTcyNDY