Speech enhancement using GAN

Bc. Matej Šutý

FIT CTU Prague
20.12.2022

I. Introduction

GAN models have been successfully used for speech enhancements and currently they are the SoA method. The CMGAN model used in this work has been recently developed in University of Stuttgart and shows promising results on benchmark data sets in English language. This work inspects the ability of the model to enhance speech in different language, namely Spanish language. I will compare the quality of speech improvement of: a) pretrained model of the authors of CMGAN, b) model trained from scratch on data set in Spanish language, c) fine-tuned model based on the pretrained model using the data set in Spanish language.

A. CMGAN model

Conformer-based MetricGAN (CMGAN) model is used for monaural speech enhancement. It consists of a generator and a metric discriminator. The generator is made of two-stage conformer blocks, a mask decoder and a complex decoder. The mask decoder estimates the mask for the input magnitude and the complex decoder estimates the compensation for real and imaginary components [1]. Transformer models are good at capturing content-based global interactions, while CNNs exploit local features effectively. Conformers combine convolution neural networks and transformers to model both local and global dependencies of an audio sequence in a parameter-efficient way [2]. The last part of the generator is the encoder, which extracts the output from the two-stage conformer blocks and it aims to predict a) mask to be multiplied by the input magnitude b) the real and imaginary parts.

The metric discriminator aims to mimic the evaluation metrics such as PESQ and STOI, which are non-differentiable. In 4 convolution blocks and global average pooling a two feed-forward layer with sigmoid activation is present. The loss function is a combination of magnitude loss and complex loss.

II. Data

The compared models are trained on both English and Spanish speaking data sets. The data are augmented with recorded noise from the DEMAND dataset [3].

A. English speaking data set

The data set used by the authors of CMGAN consists in thousands of utterances from 28 speakers and mixed with 8 background noise types. All samples are in 16kHz. More details are available in the original paper [1]. This data set was used in an extensive training of a model, which will be referred to as the *pre-trained model.*

B. Spanish speaking data set

For comparing the quality of enhancement and training the model a data set of voice in Spanish language was needed. The original data set consists of 10 speakers - 2 female a 8 male who are native Spanish speakers. The audio comes from 4 podcasts found on YouTube in high-quality (2 channels, 44kHz wav) [4] which are re-sampled to 16kHz and mono channel to follow the suit of the pre-trained model.

The noise comes from 4 sources in the DEMAND data set:

- OFFICE - a small office with a three people using computers.
- PRESTO - a university restaurant at lunchtime.
- MEETING - a meeting room while the microphone array is discussed.
- SQUARE - a public town square with many tourists.

Each noise source has 16 samples 5 minutes in duration. The noise and clean audio was combined as per following table.

The clean audio files are 5 minutes each from 4 podcasts. Each 5 minute file is split in 1 minute long files. These 5 short files are augmented with 5 different noises from same noise source so that each noise and sound has unique combination.

TABLE I
Combinations of noise and clean audio

	OFFICE	**PRESTO**	**MEETING**	**SQUARE**
Podcast 1	1-5	1-5	1-5	-
Podcast 2	6-10	6-10	6-10	-
Podcast 3	11-15	11-15	11-15	-
Podcast 4	-	-	-	1-5

Later, each augmented sound file is split using silence tokenizer from *Auditok* Python package [5] into several files between 2 and 8 seconds long.

The resulting duration of all sound files is 50 minutes. The train data set has duration 45 minutes and uses podcasts $1-3$ and noises *OFFICE, PRESTO, MEETING*. The test data set uses podcast 4 and noise *SQUARE*.

III. Training

In this work two models were trained. One was trained from scratch and the other used pre-trained model as a base for training. Hyper-parameters can be found in the Table II. Batch size 2 was selected to fit the memory constraints of computation unit.

TABLE II
HYPERPARAMETERS FOR TRAINING

	LR gen.	LR disc.	# epochs	Decay epoch
1st Scratch	5e-4	1e-3	5	-
2nd Scratch	1e-8	2e-8	17	-
3rd Scratch	1e-5	2e-5	30	5
4th Scratch	1e-8	2e-8	40	5
1st Finetuned	5e-4	1e-3	5	-
2nd Finetuned	1e-8	2e-8	50	-
3rd Finetuned	1e-7	2e-7	40	10

First attempts to train the model showed that the recommended hyper-parameters are not suitable. The discriminator loss immediately fell to almost zero while the generator loss values were very close to one.

I decided to drastically lower the learning rate to see a difference - the performance metric (PESQ) improved significantly. However, the training was unstable and no model seemed to converge as can be seen in the Figure 1.

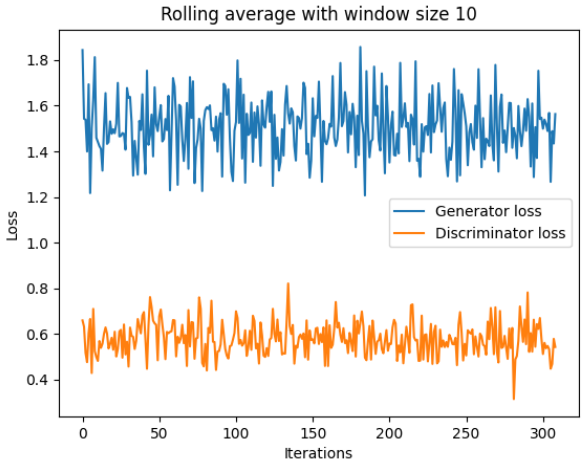

Fig. 1. Fine-tuned model with LR 1e-8.

Then I tried to use the *decay epoch* hyperparameter, which tells the model to halve the LR each X epochs. I set the parameter to 5 for training the model for scratch, because I wanted to train the model from the beginning with high LR as can be see in Table II *3rd Scratch*. This way the LR at the end of learning is $1e-7$ as can be seen in Figure 2.

However in this approach the model collapsed and the discriminator was doing too good of a job and it's loss was quickly around zero as can be seen in Figure 3. The enhanced samples of this model are just a static noise. So I went back to lower LR (4th Scratch in Table II) which produced good results, which can be heard in the AudioSamples folder.

With the fine-tuning of pretrained model the situation ended up more positive. The *decay epoch* parameter was set to 10 and the model seemed to converge with discriminator loss around 50%, which means the generator was working well and it fooled the discriminator, Figure 4. Also, the enhanced audio is clear and the background noise is removed as can be heard in the samples.

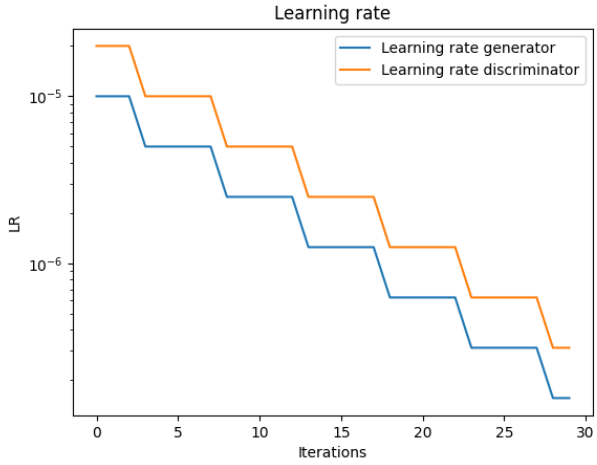

Fig. 2. LR decay every 5 epochs.

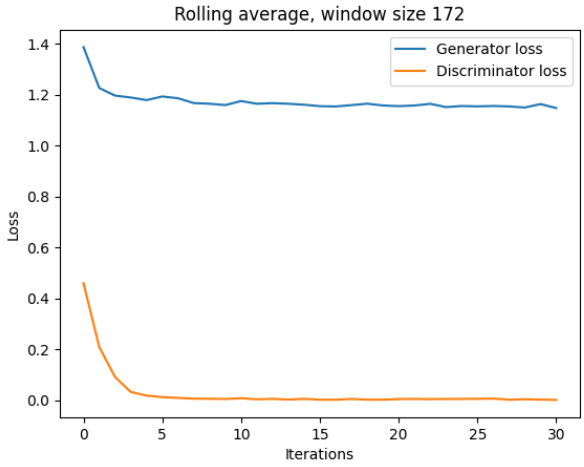

Fig. 3. Learning from scratch losses.

In both cases the windows for rolling average was set to 172 as it was the number of iterations per epoch.

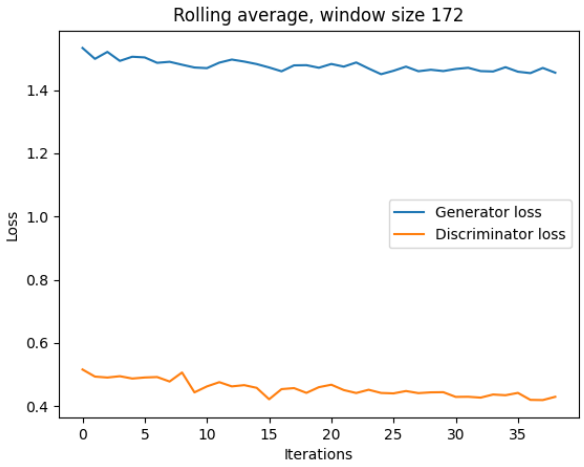

Fig. 4. Fine-tuned learning losses.

IV. Results

A. Metrics

PESQ: Perceptual Evaluation of Speech Quality, PESQ, provides a more objective, scientific approach. Otherwise known as ITU-T P.862, PESQ arrived in 2001. It analyzes the speech signal and provides an end-to-end quality. The score ranges from 1.0 up to 4.5, where values close to 4.5 indicate better speech quality, and values close to 1.0 indicate worse speech quality [6].

STOI: Short-Time Objective Intelligibility is a metric to predict intelligibility of (quite) noisy speech - not speech quality (which is typically evaluated in silence). The underlying subjective tests of this method are intelligibility tests (asking for recognized word/syllables/logatoms, etc.) [7]. The range is between 0 and 1, where higher score means higher intelligibility.

B. Discussion

The results show us that custom training and fine-tuning can give some marginal increase in quality. We should take into account that the *training* data set was rather small and that the model was trained in Google Colab notebook using T4 GPU for around 1.5 hours per 10 epochs.

To me this means that the training was not worth it as the difference between available pre-trained model and fine-tuned model is very small. Also, the model trained from scratch suffered collapse where generator was unable to learn and fool the discriminator and the produced results are useless. The following model # 4, trained with low LR, produced meaningful results but the quality was inferior to the fine-tuned model.

In the, the trained models out-performed the pre-trained model and the reader can form his own opinion by listening to the enhanced samples. I don't think that the different language influenced the results significantly.

TABLE III
Model evaluation score

	PESQ	**STOI**
Pre-trained	1.230	0.245
1st Scratch	1.064	0.317
2nd Scratch	1.103	0.254
3rd Scratch	-	-
4th Scratch	1.081	0.217
1st Finetuned	1.035	0.233
2nd Finetuned	1.234	0.245
3rd Finetuned	**1.244**	**0.247**

References

[1] Cao R., Abdulatif S., Yang B., CMGAN: Conformer-based Metric GAN for Speech Enhancement
[2] Gulati A., Qin J., et al. Conformer: Convolution-augmented Transformer for Speech Recognition
[3] C. Valentini-Botinhao, X. Wang, S. Takaki and J. Yamagishi, Investigating RNN-based speech enhancement methods for noiserobust text-to-speech.
[4] https://gitlab.fit.cvut.cz/sutymate/mvi-sp/-/blob/master/sources.txt
[5] https://auditok.readthedocs.io/en/v0.1.8/
[6] Perwej, Y., Parwej, F., Perceptual Evaluation Of Playout Buffer Algorithm For Enhancing Perceived Quality Of Voice Transmission Over Ip Network.
[7] https://stackoverflow.com/questions/60267864/pesq-stoi-score-speech-quality-for-different-languagesnon-english