COMPRESSED REPRESENTATION OF CEPSTRAL COEFFICIENTS VIA RECURRENT NEURAL NETWORKS FOR INFORMED SPEECH ENHANCEMENT

Carol Chermaz[†]*, Dario Leuchtmann*[⋆]*, Simon Tanner*[⋆]*, Roger Wattenhofer*[⋆]

[†] The Centre for Speech Technology Research, University of Edinburgh
[⋆] ETH Zurich

ABSTRACT

Speech enhancement is one of the biggest challenges in hearing prosthetics. In face-to-face communication devices have to estimate the signal of interest, but playback of speech signals from an electronic device opens up new opportunities. Audio signals can be enriched with hidden data, which can subsequently be decoded by the receiver. We investigate a hybrid strategy made of signal processing and RNN (Recurrent Neural Networks) to calculate and compress cepstral coefficients: these are descriptors of the speech signal, which can be embedded in the signal itself and used at the receiver's end to perform an Informed Speech Enhancement. Objective evaluations showed an increase in speech quality for noisy signals enhanced with our method.

Index Terms— Speech enhancement, Cepstral Smoothing, Recurrent Neural Networks

1. INTRODUCTION

Separating speech from a noisy mixture is a widely studied problem in the hearing prosthetics domain, as speech intelligibility is compromised by the presence of noise. Modern prosthetic devices run several algorithms designed to tackle this issue, from beamformers – which take advantage of multiple microphones – to statistical signal estimators like Wiener filters. When the signal of interest is unknown, the problem can also be described as *Blind Source Separation*.

A different scenario is posed when there is some prior knowledge about the signal of interest, and the problem can be categorized instead as ISS (Informed Source Separation) [1]. In recent decades different methods have been proposed in this field, mainly in the context of music remixing or *active listening* (e.g., a scenario in which the listener can choose which sound source to enhance in respect to the mixture). The system proposed in [2] utilizes information from a music score in order to separate instruments acoustically from a monophonic mixture; in [3] information about the different sources is encoded in the signal itself by quantizing the MDCT (Modified Discrete Cosine Transform) coefficients; source and receiver need a common *code-book* to operate, and the method can be used for active listening.

An application that is still unexplored is the embedding of data in speech signals for the purpose of being decoded by hearing devices and used to perform *Informed Speech Enhancement* when exposed to playback through the acoustic channel.

In [4] a method for embedding text into speech signals is described, while in [5] textual information is used to enhance speech from a noisy mixture by utilizing some features of TTS (Text To Speech), given that a transcription of the utterances is available. While a combination of these methods seem to point in the direction of our intended application, we are looking to represent the signal with acoustic descriptors rather than text to then enhance the speech signal.

While being very efficient at source separation, the ISS method described in [3] utilizes a high capacity (up to 150 kbps in music signals) watermarking technique based on [6] which is not intended for acoustic propagation. Its use-case is in fact for the watermark to be read by a device from the digital file directly, whereas acoustic propagation would destroy it. Watermarking techniques that are inaudible and robust to acoustic propagation have been investigated in the context of *second screen* applications; however, such methods are subject to a heavy trade-off between capacity and robustness to the acoustic path (e.g., up to 10 bps in [7]). The OFDM technique proposed in [8] is designed for acoustic propagation and has been shown to achieve data rates up to 400 bps; however, it is designed to be hidden within music signals, as the richness in frequency content allows for perceptual masking of high frequency carriers. Such a method is therefore less suitable to be used in "bare" speech signals.

With the described application in mind – a hearing device detecting a watermark from speech playback through the acoustic channel – we chose a descriptor of the speech signal that requires a limited amount of data, while still being useful for signal separation: its *cepstral coefficients* [9]. Only a small number of such coefficients is necessary to reconstruct the spectral envelope of the signal (the higher the number

The authors of this paper are alphabetically ordered.

This project has received funding from the EU's H2020 research and innovation programme under the MSCA GA 675324 (the ENRICH network: www.enrich-etn.eu). We would like to thank Simon King[†] for the idea of embedding information in the speech signal for speech enhancement.

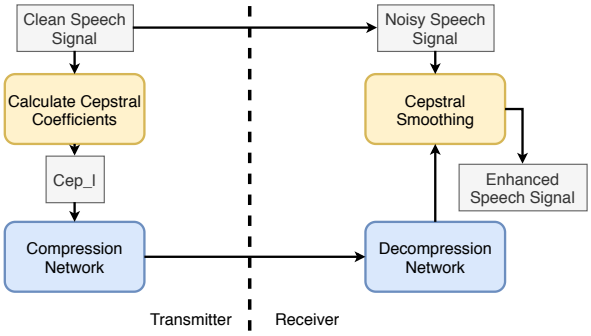

Fig. 1. System overview of the trained neural networks. At the transmitter, the compression of the cepstral coefficients is performed while the receiver does the decompression and cepstral smoothing of the received noisy signal.

of coefficients, the greater the definition). Such coefficients are widely used in ASR (Automatic Speech Recognition) and TTS (Text to Speech) applications. In our intended application scenario, coefficients could be sent to the receiver and used to obtain the spectral envelope of the signal via IFFT (Inverse Fast Fourier Transform), in order to filter out the ambient noise via spectral subtraction. This system can be applied in any scenario where speech signals are broadcast from loudspeakers, such as public speeches, announcements in train stations, etc. The transmitter does not need any additional hardware, as the data can be embedded directly in the audio.

The use of cepstral coefficients for the purpose of speech enhancement is a known technique [10, 11] also referred to as *cepstral smoothing* [12] when used in order to prevent *musical noise* in addition to other noise reduction strategies. In this study, we use it as the sole method of enhancement.
Even though cepstral coefficients may offer a compact acoustic description the signal, the amount of data required for an accurate representation is still too large for any known acoustic watermarking technique (e.g. 156 floating point numbers per second = 4992 bps). For this reason, we try to compress this representation even further with a machine learning method.

2. METHODS

We calculate the cepstral coefficients by dividing the original speech signal (unaffected by noise) in short Hann-windowed time frames. For each frame, coefficients are obtained with:

$$F[log|F(W_k)|] \tag{1}$$

Where W_k is the *k-th* time frame, and $F(.)$ represents the FFT (Fast Fourier Transform). We then select only the first few coefficients.

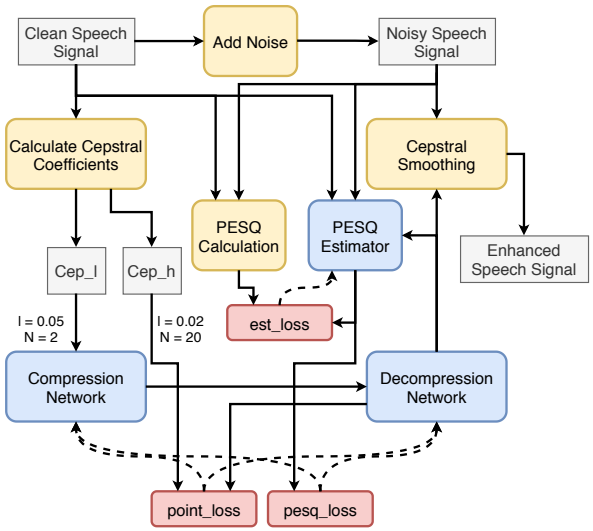

Fig. 2. System overview of the neural networks used to train the transmitting compression network and the receiving decompression network. The solid lines indicate the data flow while the dashed ones indicate the back-propagation of the loss. The gray boxes represent data, the red ones loss functions, the blue ones are the neural networks, and the yellow ones are classic functions which are not differentiable.

We have developed two neural networks; one to compress the cepstral coefficients at the transmitter and a second one to reconstruct the data at the receiver.

The network located at the transmitter is a convolutional recurrent neural network (CRN) which compresses the cepstral coefficients gathered from the clean speech signal into eight floating point numbers per second. For this task we use a CRN because of its ability to store data in a Long-Short-Term-Memory, which allows to detect time-dependent characteristics of the cepstral coefficients. At the receiver, a linear neural network (LNN) decompresses the values back to the cepstral coefficients. This compression scheme is lossy, the coefficients are not restored perfectly.

At the transmitter the cepstral coefficients are calculated with overlapping windows with a length of $l = 0.05$ seconds. For each window only the first two coefficients are used. This results in 156 coefficient with a total of 4992 bits per second. These coefficients are then compressed with the `Compression Network` to 256 bits per second. The `Decompression Network` then decompresses this data to 20 cepstral coefficients per window of 0.02 seconds. The compression and decompression are not symmetric, after the decompression there is more data than before the compression.

Figure 1 shows an overview of the running system.[1]

[1]Samples can be found at `https://drive.google.com/drive/folders/1MKcdpS01wY7N5luzw0hf9Ek96xKyTeAj`

2.1. Quality Estimation

A well suited loss function is crucial to achieve good results with a neural network. The most obvious loss function is the mean squared error of the decompressed cepstral coefficients compared to the real cepstral coefficients, which results in the `point_loss`. While this guarantees that the decompressed data is similar to the target data, it does not guarantee that the decompressed cepstral coefficients are suitable for speech enhancement via cepstral smoothing. The quality of the coefficients can only be measured by judging the intelligibility and quality of the enhanced speech signal.

In order to evaluate our output we chose to use PESQ [13]. Although being primarily an objective measure of speech quality, it has been found to correlate well also with intelligibility [14]. Its score ranges from -0.5 (bad) to 4.5 (excellent).

In our case, the closer the noisy signal after the cepstral smoothing is to a perfectly clean one, the better is the PESQ value. For this reason the difference between the perfect PESQ value and the one of the enhanced speech file is a well suited loss function. Unfortunately, the computation of the PESQ value is not differentiable and can therefore not be used directly as a loss function in a neural network. Furthermore in our case the output of the decompression network are the cepstral coefficients which first have to be used for the cepstral smoothing. These computations are also not differentiable.

To circumvent these issues we have built a third neural network, the `PESQ Estimator`, with eight fully connected linear layers. This network takes the decompressed cepstral coefficients and the noisy and clean speech signals as inputs and estimates the PESQ value. The mean squared error of this estimated PESQ value to the perfect PESQ value of 4.5 is then used as `pesq_loss` in the loss function for the compression and decompression network. Figure 2 shows the whole system during training where also the PESQ estimator is included. The PESQ estimator is trained after each training round of the compression and decompression networks. The estimated PESQ value is compared to the real PESQ value. The resulting `est_loss`, which is the mean squared error, is then back propagated to the PESQ estimator.

We combine the `point_loss` and the `pesq_loss` as our loss function for the compression and decompression networks. This guarantees that the networks search for cepstral coefficients that minimize the mean squared error compared to the original ones while the quality of the speech signal is maximized.

2.2. Compression Network

For the compression of the cepstral coefficients, a convolutional recurrent neural network (CRN) is used. CRNs have a long short-term memory (LSTM) that allows to pass on information processed in the past to the current processing step. The speech signal at a time t is influenced by the signal at time $t-1$. LSTMs are perfectly suited to model this causal influence on the output. Our CRN consists of three encoder layers, a long-short term memory layer in the middle and three decoder layers to detect the time properties of the cepstral coefficients. At the end we have four fully connected linear layers that compress the detected properties to eight floating point coefficients which are then transmitted to the receiver.

We have developed our CRN motivated by recent works of Ke Tan and DeLiang Wang in [15]. Our network allows real time speech enhancement since it only considers causal data.

While in [15] the network uses the noisy audio data as an input to enhance it, we use the cepstral coefficients calculated before from the clean speech signal. We have therefore adapted the original network to our smaller input size. But even this downsized network from [15] is far too big for a typical hearing aid and requires too much computational power; for this reason, we use this network only on the transmitter where neither computational power nor storage is a limiting factor.

2.3. Decompression Network

At the receiver the noisy speech signal should be enhanced by filtering out the background noise, while the speech signal remains clear and therefore more intelligible. The data from the transmitter CRN is used as input to the decompression network. The network outputs $N = 20$ cepstral coefficients for each window of $l = 0.02$ seconds. These coefficients can then be used for cepstral smoothing of the speech signal.

The decompression network is a linear neural network (LNN). An LNN consists only of fully connected linear layers. While CRNs with their LSTM are capable of using information from the past, an LNN is typically simpler and therefore requires much less computational power and data storage. This is of great importance at the receiver. The decompression network consists of nine linear layers and only requires 38 KB of storage.

2.4. Data

The training of the proposed neural networks is performed using the data from [16]. There are 11572 speech files spoken by 56 different speakers for the training and additional 824 speech files for testing. From these samples we generate the noisy samples for the receiver. We add echoes and additive noise of people talking to simulate a realistic environment. From each sample a one second long interval is chosen randomly and used for training. After every five epochs the training data gets regenerated and new one second long intervals are chosen from each speech file.

In Figure 1 the system after the training is depicted. The PESQ estimator is not used anymore.

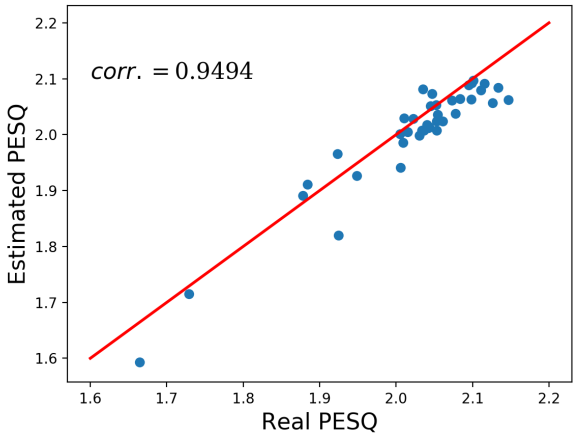

Fig. 3. In this graph the correlation between the estimated and real PESQ value is depicted. The red line indicates a perfect correlation of 1.0 while the blue dots represent the effective results from the measurement.

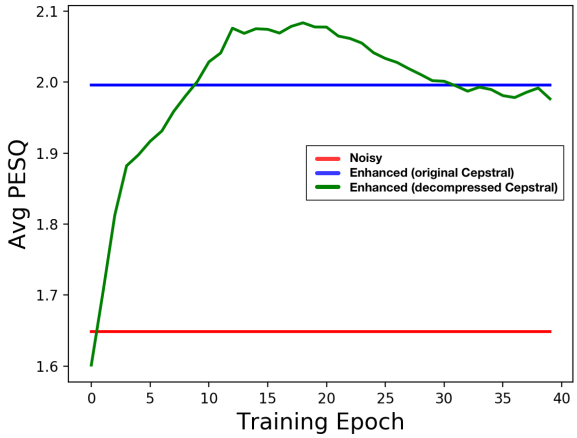

Fig. 4. The average real PESQ value measured with the testset of 824 samples per training epoch (green). The maximum is achieved in epoch 18 with an average PESQ value of 2.08. The blue and red lines indicate the average real PESQ value of the noisy audio and the audio enhanced with the original cepstral coefficients respectively.

3. RESULTS

3.1. PESQ Estimation

The PESQ estimation is a crucial part of the system. The quality of our loss function directly depends on the quality of the PESQ estimation. Figure 3 shows the estimated and real average PESQ values for 150 random test samples for each epoch. As it can be seen, the estimated PESQ value clearly follows the real PESQ value. In fact the correlation between these two is with 0.95 almost perfect.

This allows us to use this PESQ estimator in our system to improve the quality of our loss function for the compression and decompression networks.

As it can be seen in Figure 4 the average PESQ value has a value between 1.9 and 2.2 after the first few epochs. With a variance of roughly 0.3 about 70 % of all the estimated PESQ values are in the range between 1.35 and 2.75. This leads to a mean squared error between 3 and 10 for the `pesq_loss`.

3.2. Compression and Decompression Networks

The goal of the neural networks is to find well suited cepstral coefficients with only very little input data at the decompression network.

Figure 4 shows the comparison of using the original and the decompressed cepstral coefficients for the cepstral smoothing of the noisy speech signal. In the first few epochs, the learning curve is quite steep and the network produces already good results. From epoch 12 to 20 the network is still slowly improving but not a lot anymore and after epoch 20 overfitting occurs. The maximum is achieved in epoch 18 with an average real PESQ value of 2.08. This is more than the improvement with the original unprocessed cepstral coefficients. Our network is therefore capable to detect some features from the cepstral coefficients such that the compression and decompression leads to a slight improvement of the enhancement.

We have shown that the PESQ value is strongly improved with the help of our networks while we were able to reduce the transmitted bits per second to only 256.

4. CONCLUSION

We have proposed a method for informed speech enhancement based on cepstral smoothing. While the cepstral coefficients are too large to be directly embedded in the speech signal via acoustic watermarking, we have developed a neural network to compress them before transmission. At the receiver they are decompressed using only a small neural network. From originally 156 floating point numbers per second we can drastically reduce the amount of data to 8 floating point numbers. Our results show that the speech quality of the output is even improved slightly compared to directly using the original, uncompressed cepstral coefficients for cepstral smoothing.

While our results are promising in terms of speech enhancement, the amount of data required for acoustic transmission is still too large for known methods. Future work could investigate a high-capacity watermarking technique which satisfies the requirements of robustness and inaudibility.

5. REFERENCES

[1] Kevin H Knuth, "Informed source separation: A bayesian tutorial," in *2005 13th European Signal Processing Conference*. IEEE, 2005, pp. 1–8.

[2] Zhiyao Duan and Bryan Pardo, "Soundprism: An online system for score-informed source separation of music audio," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 6, pp. 1205–1215, 2011.

[3] Mathieu Parvaix, Laurent Girin, and Jean-Marc Brossier, "A watermarking-based method for informed source separation of audio signals with a single sensor," *IEEE Transactions on audio, speech, and language processing*, vol. 18, no. 6, pp. 1464–1475, 2009.

[4] Pramatha Nath Basu and Tanmay Bhowmik, "On embedding of text in audio a case of steganography," in *2010 International Conference on Recent Trends in Information, Telecommunication and Computing*. IEEE, 2010, pp. 203–206.

[5] Keisuke Kinoshita, Marc Delcroix, Atsunori Ogawa, and Tomohiro Nakatani, "Text-informed speech enhancement with deep neural networks," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[6] Brian Chen and Gregory W Wornell, "Quantization index modulation: A class of provably good methods for digital watermarking and information embedding," *IEEE Transactions on Information theory*, vol. 47, no. 4, pp. 1423–1443, 2001.

[7] Michael Arnold, Xiao-Ming Chen, Peter Baum, Ulrich Gries, and Gwenael Doerr, "A phase-based audio watermarking system robust to acoustic path propagation," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 3, pp. 411–425, 2013.

[8] Manuel Eichelberger, Simon Tanner, Gabriel Voirol, and Roger Wattenhofer, "Imperceptible audio communication," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 680–684.

[9] Sirko Molau, Michael Pitz, Ralf Schluter, and Hermann Ney, "Computing mel-frequency cepstral coefficients on the power spectrum," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221)*. IEEE, 2001, vol. 1, pp. 73–76.

[10] Hadi Veisi and Hossein Sameti, "Speech enhancement using hidden markov models in mel-frequency domain," *Speech Communication*, vol. 55, no. 2, pp. 205–220, 2013.

[11] Duncan Bees, Maier Blostein, and Peter Kabal, "Reverberant speech enhancement using cepstral processing," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on*. IEEE Computer Society, 1991, pp. 977–980.

[12] Colin Breithaupt, Timo Gerkmann, and Rainer Martin, "Cepstral smoothing of spectral filter gains for speech enhancement without musical noise," *IEEE Signal processing letters*, vol. 14, no. 12, pp. 1036–1039, 2007.

[13] ITU, "P.862 : Perceptual evaluation of speech quality (pesq): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs," 2008, `https://www.itu.int/rec/T-REC-P.862/en`, accessed 2020-10-21.

[14] John G Beerends, Erik Larsen, Nandini Iyer, and J Van Vugt, "Measurement of speech intelligibility based on the pesq approach," in *Proceedings of the Workshop Measurement of Speech and Audio Quality in Networks (MESAQIN), Prague, Czech Republic*, 2004.

[15] Ke Tan and DeLiang Wang, "A convolutional recurrent neural network for real-time speech enhancement," in *Proc. Interspeech 2018*, 2018, pp. 3229–3233.

[16] University of Edinburgh, "Noisy speech database for training speech enhancement algorithms and tts models," 2017, `https://datashare.is.ed.ac.uk/handle/10283/2791`, accessed 2020-10-21.