Spectral Envelope Estimation using Deep Recurrent Neural Networks for Speech Reconstruction

Stefan Ciba, Mohammed Krini
stefan.ciba@th-ab.de, mohammed.krini@th-ab.de
Aschaffenburg University of Applied Sciences, Laboratory for Signal Processing

Abstract
Classical algorithms for speech enhancement mostly consist of digital filters, which utilize a statistical approach for optimal estimation of the noise power spectrum. Because of constraints classical algorithms can lead to unsatisfactory results, especially at high noise. Machine learning is used for the challenging task of speech enhancement and particularly in reconstruction of speech components that are masked by noise. For speech reconstruction the source-filter model of speech production is applied. The model can be used for speech reconstruction in such a way, that the excitation signal (produced by the human glottis), as well as the so-called envelope (filter characteristics of the vocal tract), are estimated separately and subsequently combined. In this work a deep recurrent neural network (deep-RNN) is applied as regressive model to estimate an IIR-Filtered spectral envelope in the first approach, as well as a clean cepstral envelope representation in a second approach. Finally, the proposed approaches are analyzed objectively and compared.

Introduction
Speech enhancement can be done by applying analytical models such as the source-filter model of speech production [1]. Together with data driven estimation techniques it is possible to recover almost lost speech signal parts, even at high noise, which can be denoted as speech reconstruction [2].
Since the envelope is a important part of a speech signal this work focuses on that, without attempting to be a full speech enhancement. In the same manner previous work focused on using repetitive *Long Short-Term Memory* (LSTM) layers for narrow-band spectral envelope estimation (up to 2 kHz), which emerges from an IIR-Filter [3][4][5]. Further motivation of this work is to estimate an entire broadband envelope (up to 8 kHz) after hyperparameter tuning and extending formerly used training methods. Since in cepstral domain a broadband envelope representation requires only a small number of coefficients, the hypothesis is that size of the neural network can be reduced by a Cepstral-Lifter approach. Estimation of cepstral coefficients has shown significant improvement of speech quality [6].
The *Log Spectral Distance* (LSD) [7] is used as objective measure to compare the spectral envelopes for speech-frames, as well as for noise-frames. Finally the best found hyperparameter sets for both feature extraction approaches are discussed concerning size of the neural network and speech intelligibility (STOI) [8].

Speech Feature Extraction
Assuming a single channel scenario, the time discrete microphone signal y_n can be described as an addition of the desired speech signal s_n and noise d_n as follows:

$$y_n = s_n + d_n. \tag{1}$$

The source-filter model of speech production is a LTI system approach to classify two subsystems of a human speech signal production [7] wherein:

a) The filter denotes the envelope, which is produced by the vocal tract.

b) The source denotes the excitation, which is produced by the human glottis.

It is applied introducing the speech s_n as a convolution of the envelope a_n, with the excitation b_n:

$$s_n = a_n * b_n. \tag{2}$$

Analysis filterbank
A windowed STFT is performed on the discrete time domain expression 1 by DFT-modulated filterbank [9] using an analysis window $h_{\mathrm{ana},k}$ (Slepian), a frame width of $N_{\mathrm{DFT}} = 512$ where μ is called subband index counting $\mu = 0, ..., N_{\mathrm{DFT}}$ frequency bins. The window applies a frame shift over $r = 128$ discrete time steps, at a sampling frequency of $f_s = 16$ kHz, with $\Omega_\mu = \frac{2\pi\mu}{N_{\mathrm{DFT}}}$ as follows:

$$Y_{\mu,\eta} = \sum_{k=0}^{N_{\mathrm{DFT}}-1} y_{\eta r-k} h_{\mathrm{ana},k} e^{-j\Omega_\mu k}, \tag{3}$$

whereas η represents the frame index. The conjugate complex spectral components get neglected since redundancy isn't necessary for further processing except back transformation, so that a upper boundary $M = \frac{N_{\mathrm{DFT}}}{2} + 1$ can be used meanwhile. However, only the magnitudes of the complex spectrum vectors $Y_{\mu,\eta}$ are used for the envelope estimation.

Spectral envelope extraction by IIR-Filter
The envelope $A_{\mu,\eta}$ in spectral domain is extracted by means of a first order IIR-Filter approach applied on magnitude spectrum $Y_{\mu,\eta}$. The implementation consists of a forward and backward smoothing with a smoothing constant $\lambda = 0.8$ of each STFT frame (low-pass) [5]. The forward smoothing is done recursively [10] as follows:

$$Y'_{\mu,\eta} = \begin{cases} |Y_{\mu,\eta}|, & \text{if } \mu = 0, \\ \lambda Y'_{\mu-1,\eta} + (1-\lambda)|Y_{\mu,\eta}|, & \text{else.} \end{cases} \tag{4}$$

The backward smoothing is done analogues yielding the envelopes $A^y_{\mu,\eta} = Y''_{\mu,\eta}$.
In addition the excitations are extracted by element-wise division: $B^y_{\mu,\eta} = \frac{Y_{\mu,\eta}}{A^y_{\mu,\eta}}$. The reconstruction takes place by combining the excitation and the estimated envelope again by element-wise multiplication: $\hat{Y}_{\mu,\eta} = B^y_{\mu,\eta}\hat{A}^y_{\mu,\eta}$.

Cepstral envelope representation
The real cepstral coefficients [11] based on the STFT in Eq. 3 are calculated by

$$C^y_{\mu,\eta} = \frac{1}{N_{\mathrm{DFT}}} \sum_{k=0}^{N_{\mathrm{DFT}}-1} 20\log_{10}|Y_{\mu,\eta}|e^{j\Omega_\mu k}. \qquad (5)$$

A lifter is applied by splitting the vectors of Eq. 5 by μ. The lower $\Gamma \approx 20$ coefficients of Eq. 5, counting $\mu = 0, ..., \Gamma - 1$, represent the envelope $A^y_{\mu,\eta}$, used by the RNN. The excitation $B^s_{\mu,\eta}$ is obtained by neglecting the first Γ coefficients so that the index counts $\mu = \Gamma, ..., M$. The reconstruction is done by joining the vectors and mirror to a full cepstrum that consists of the estimated envelope representation by the RNN and the clean excitation. A DFT and inversion of the logarithm is used for back transformation into spectral domain.

Feature normalization
The Eq. 6 shows how the normalization [12] is applied. It is done by mean subtraction followed by division of the standard deviation sigma over each μ feature, along every $\eta = 0, ..., N$ time-frame of a whole data set:

$$\tilde{A}_{\mu,\eta} = \frac{A_{\mu,\eta} - \overbrace{\frac{1}{N}\sum_{\eta=0}^{N-1} A_{\mu,\eta}}^{\text{mean: } \bar{A}_\mu}}{\underbrace{\sqrt{\frac{1}{N}\sum_{\eta=0}^{N-1}\left[A_{\mu,\eta} - \frac{1}{N}\sum_{\eta=0}^{N-1} A_{\mu,\eta}\right]^2}}_{\text{sigma: } \sigma_{A_\mu}}}. \qquad (6)$$

The normalization paradigm by choice is to normalize clean speech features first and applying the obtained parameters on the noisy speech features. The same normalization method is applied to the IIR-Filtered spectral envelopes as well as on the cepstral representation of the envelopes. Denormalization is done vise versa by multiplication of sigma and adding mean.

RNN Architecture

The implementation of the well known Deep-RNN is done as shown in [12] with repetitive layers that consist of LSTM-cells and the *Leaky ReLU* activation function. The number of layers is defined by n_{layer} and the number of LSTM cells per layer is defined by n_{neurons}. The size of the neural net can be calculated by $n_{\mathrm{size}} = n_{\mathrm{layer}} \cdot n_{\mathrm{neurons}}$.

Data set
The data set consists of 236 speech samples of one and the same female speaker chosen from the *Edinburgh Corpus* [14]. Noisy speech data is produced based on clean data by adding car interior noise at a driving speed of 140 km/h for several *Signal-to-Noise Ratios* [7] (SNR from −5 dB to 15 dB in 5 dB steps) by means of Eq. 1. The noise signal has its significant power in the lower half frequency band as usual for interior noise of a driving car [2]. This minimalistic setup has been chosen for the purpose of architecture tuning and finding good training parameters. The data set is split as follows: Training (85 %), validation (13 %) and application (2 %).

Optimization
The *Adam Optimizer* of Tensorflow is used to minimize the *Mean Squared Error Loss* (MSE) [12] and is applied every epoch on a number of feature-frames N_{mb} which is denoted as the mini-batch-size. A batch however represents the whole data set which is entirely trained during one epoch by *Mini-Batch Gradient Descent* [12] training method. The MSE gets calculated for normalized clean envelopes $\tilde{A}^s_{\mu,\eta}$ and normalized estimated envelopes $\breve{A}^y_{\mu,\eta}$, as follows:

$$L_{\mathrm{MSE}} = \frac{1}{N_{\mathrm{mb}}}\frac{1}{M}\sum_{\eta=0}^{N_{\mathrm{mb}}}\sum_{\mu=0}^{M-1}\left(\tilde{A}^s_{\mu,\eta} - \breve{A}^y_{\mu,\eta}\right)^2. \qquad (7)$$

During training the loss gets double checked by validation by applying same model parameters on the validation data set. If validation loss gets less, the model parameters gets saved, otherwise the model has become worse and the learning rate r_{learn} is decreased by $\frac{1}{2}$, which is also known as *Performance Scheduling* [12]. If validation loss gets much bigger or much smaller than training loss, the training runs into overfitting or underfitting, respectively and thus training stops at this point to save computation time, which is referred to as *Early Stopping* [12]. *Dropout* [13] as well as ℓ_2-*Regularization* [12] are used to reach less overfitting and less complexity during training.

Evaluation Methods

Speech activity detection
In order to apply measures separately, a speech activity detection for binary frame masking is realized to distinguish speech-frames and noise-frames by signal-to-noise ratio $\overline{\mathrm{SNR}}_\eta$ of the current frame:

$$b_\eta = \begin{cases} 1 & \text{for } \overline{\mathrm{SNR}}_\eta > \vartheta, \\ 0 & \text{else,} \end{cases} \qquad (8)$$

whereas for the noise power estimation the *Improved Minima Controlled Recursive Averaging Algorithm (IMCRA)* is used [15]. ϑ is set to 0.4. Speech-frames are selected by $\beta^{c=\mathrm{speech}}_\eta = b_\eta$ and noise-frames are selected by $\beta^{c=\mathrm{noise}}_\eta = 1 - b_\eta$. N_{f} is the number of all time-frames of same SNR, separated for each case c by $N^c_{\mathrm{f}} = \sum_{\eta=0}^{N_{\mathrm{f}}-1}\beta^c_\eta$.

Objective measure
The *Log Spectral Distance (LSD)* [7] between a set of clean and estimated envelopes $A^s_{\mu,\eta}$ and $\hat{A}^y_{\mu,\eta}$ in spectral domain gets calculated for both cases c over N_{f} as follows:

$$LSD^c = \frac{1}{N^c_{\mathrm{f}}}\sum_{\eta=0}^{N_{\mathrm{f}}-1}\beta^c_\eta\sqrt{\frac{1}{M}\sum_{\mu=0}^{M-1}\left[10\log_{10}\left(\frac{A^s_{\mu,\eta}}{\hat{A}^y_{\mu,\eta}}\right)\right]^2}. \qquad (9)$$

Results

Several parameter studies have been done in order to find a possibly optimal hyperparameter set for each feature type as shown in Fig. 1. The found sets which behave best for speech are shown in Tab. 1.

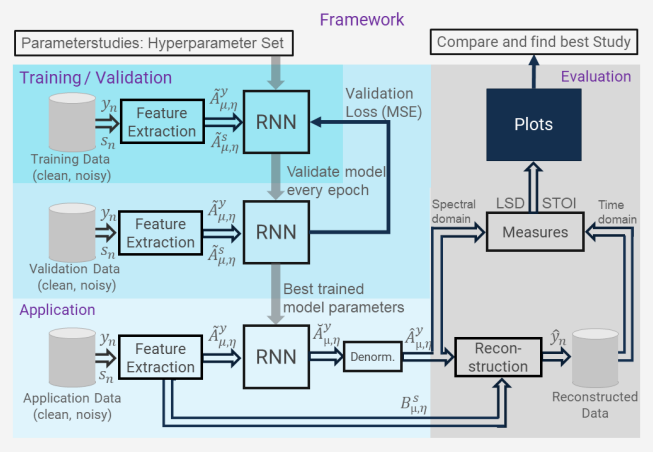


Figure 1: Block diagram of the Framework.

Table 1: Best parameters for used dataset.

Parameter	Approach	
	IIR-Filter	Cepstral-Lifter
n_{neurons}	257	32
n_{layer}	3	4
n_{size}	771	128
N_{mb}	512	2048
ℓ_2	0.001	0.001
dropout	0.4	0.2
best epoch	14	505
r_{learn}	$5e^{-4}$	$2.5e^{-5}$

Due to a reference curve (LSDs between noisy and clean envelopes), it has shown for cepstral-Lifter approach to have about 0.56 to 1.10 dB higher LSD on the used data set than the IIR-Filter approach, which has to be respected for a direct comparison of absolute values.

Increasing the number of neurons n_{neur} from 32 to 257 yield a decrease of LSD in speech regions, but vise versa in noise regions as shown in Fig. $2(a)$ and $3(a)$. The same can be said for increasing the number of layers n_{layer} from 2 to 3. The estimated envelopes in speech regions are getting significantly better from -5 dB until approximately 12 dB SNR-level, where reference and estimation curve subtend. In noise regions the estimated envelopes show improvement at any SNR level used.

Increasing the number of layers n_{layer} result in a lower overall LSD when cepstral coefficients being estimated in speech-frames as expected, but not in noise-frames, as shown in Fig. $2(b)$ and $3(b)$. The overall characteristic of LSD show unexpected behavior and rises at high SNRs. The estimated envelopes getting significantly better in speech regions until about 9 dB SNR, where reference and estimation curve subtend. In noise regions the estimated envelope of the best found parameter-set show improvement at any SNR level.

A more efficient comparison can be done by looking at the differences Δ of average LSD and STOI measures of the estimations regarding noisy- / clean-reference curves, along the SNRs, as shown in Tab. 2. The mean improvement of LSD is higher using IIR-Filter approach during speech and even more during noise and STOI shows a higher improvement, too.

Table 2: Mean improvement of RNN envelope estimations.

Measure	Approach	
	IIR-Filter	Cepstral-Lifter
ΔSTOI	26.39	6.77
$\Delta\text{LSD}^{\text{speech}}$	3.89 dB	3.26 dB
$\Delta\text{LSD}^{\text{noise}}$	22.90 dB	5.75 dB

Conclusion

The shown results lead to the conclusion that the spectral IIR-Filter approach is to be preferred over the Cepstral-Lifter approach regarding the architecture used. The IIR-Filter approach yield higher overall improvements over the Cepstral-Lifter approach but trade-off is the 6 times bigger size n_{size} regarding the two best architectures discussed. The Cepstral-Lifter approach is still an interesting research topic, which may need a different setup to perform better. Thus in this work noise- and speaker-generalization has not been taken into account yet, methodical extensions will be made in the future to prevent effects of speaker- and noise-overfitting, by more generalized data sets.

References

[1] S. Elshamy: Speech Enhancement Exploiting the Source-Filter Model. PhD-Thesis, Braunschweig, 2020.

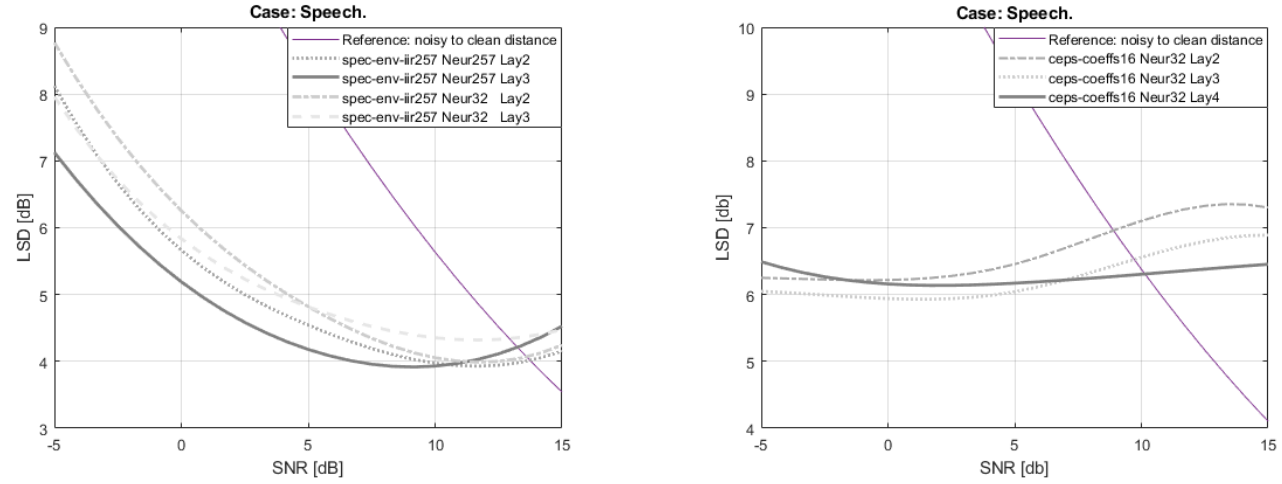
[2] M. Krini and G. Schmidt: Model-based Speech Enhancement. In E. Hänsler, G. Schmidt (eds.), Speech and Audio Processing in Adverse Environments, Springer, pp. 89-134, 2008.

[3] M. Krini and T. Stegmann: Speech Enhancement with Partial Signal Reconstruction Based on Deep Recurrent Neural Networks and Pitch-Specific Codebooks. Proceedings of ICSP 2020.

[4] C. Seitz and M. Krini: DNNs versus Codebooks based Spectral Envelope Estimation for Partial Reconstruction of Speech Signals. Workshop on Speech and Audio in the Northeast 2019 (SANE), New York.

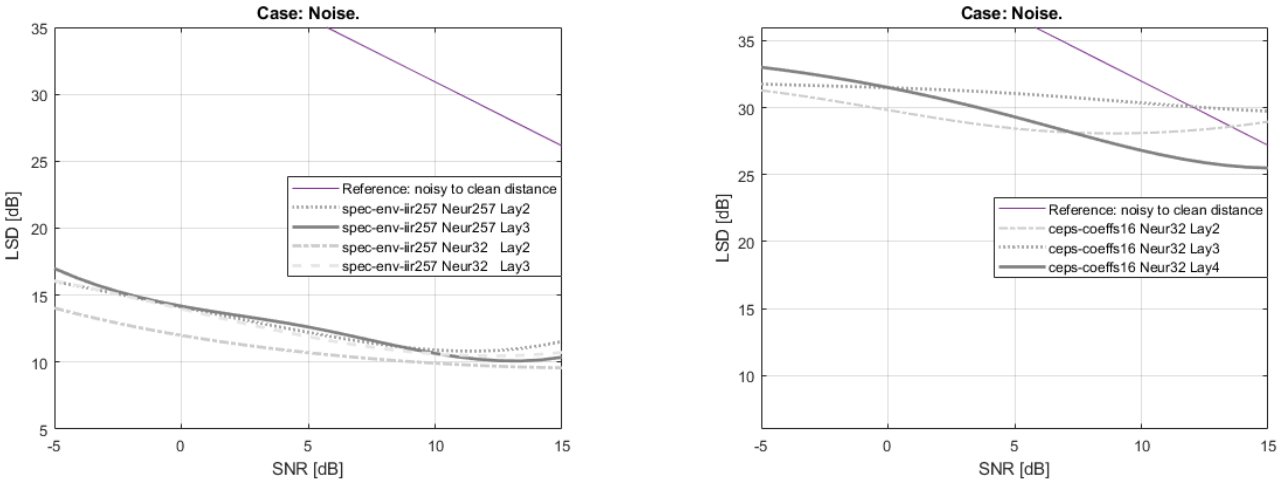
[5] C. Seitz and M. Krini: Spectral Envelope Estimation Based on Deep Neural Networks and its use for Speech Reconstruction. Proceedings of 45th German Annual Conference on Acoustics 2019 (DAGA).

[6] S. Elshamy, N. Madhu , W. Tirry and T. Fingscheidt: DNN-Supported Speech Enhancement with Cepstral Estimation of Both Excitation and Envelope. IEEE/ACM Transactions on Audio, Speech and



(a) Spectral IIR-Filter approach. (b) Cepstral-Lifter approach.

Figure 2: LSD measures for envelope estimations within different hyperparameter sets during speech.



(a) Spectral IIR-Filter approach. (b) Cepstral-Lifter approach.

Figure 3: LSD measures for envelope estimations within different hyperparameter sets during noise.

Language Processing, vol. 26, no. 12, pp. 2460 - 2474, 2018.

[7] P. Vary, U. Heute and W. Hess: Digitale Sprachsignalverarbeitung. ISBN: 3-519-06165-1, Teubner Stuttgart, 1998.

[8] C. H. Taal, R. C. Hendriks, R. Heusdens and J. Jensen: An Algorithm for Intelligibility Prediction of Time-Frequency Weighted Noisy Speech. In IEEE Transactions on Audio, Speech, and Language Processing, vol. 19, no. 7, pp. 2125-2136, Sept. 2011.

[9] E. Hänsler and G. Schmidt: Acoustic Echo and Noise Control - A Practical Approach. John Wiley & Sons, Hoboken, NJ, USA, 2004.

[10] B. Meffert and O. Hochmuth: Werkzeuge der Signalverarbeitung. Grundlagen, Anwendungsbeispiele, Übungsaufgaben. ISBN: 3-8273-7065-5, Pearson, 2004.

[11] D. G. Childers, D. P. Skinner and R. C. Kemerait: The cepstrum: A guide to processing. In Proceedings of the IEEE, vol. 65, no. 10, pp. 1428-1443, Oct. 1977.

[12] A. Géron: Hands-On Machine Learning with Scikit-learn and Tensorflow. O'Reilly Media, 2017.

[13] Y. Gal and Z. Ghahramani: A theoretically grounded application of dropout in recurrent neural network. ArXiv e-prints, arXiv:1512.05287, 2015.

[14] C. Valentini-Botinhao: Noisy speech database for training speech enhancement algorithms and TTS models, 2016 [sound]. University of Edinburgh. School of Informatics. Centre for Speech Technology Research (CSTR). URL: `http://dx.doi.org/10.7488/ds/2117`.

[15] I. Cohen: Noise Spectrum Estimation in Adverse Environments: Improved Minima Controlled Recursive Averaging. IEEE, Transactions on Speech and Audio Processing, vol. 11, no. 5, pp. 466 - 475, 2003.