Artificial Bandwidth Extension of Speech Signals using Neural Networks

Dissertation

zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften
(Dr.-Ing.)
der Technischen Fakultät
der Christian-Albrechts-Universität zu Kiel

vorgelegt von

Jonas Jungclaussen

Ulm 2021

Datum der Einreichung: 15.09.2020
Datum der mündlichen Prüfung: 18.01.2021

1. Berichterstatter: Prof. Dr.-Ing. Gerhard Schmidt
2. Berichterstatter: Univ.-Prof. Dr.-Ing. Peter Jax

Acknowledgments

This dissertation emerged during my time as external doctoral student of the Institute for Digital Signal Processing and System Theory at Kiel University that I spent at Nuance Communications Deutschland GmbH and later Cerence GmbH at the Ulm site.

First of all, I would like to thank Prof. Dr.-Ing. Gerhard Schmidt, who supervised the doctorate in a great way by leaving me free space to take my decisions and at the same time giving me good feedback whenever it was necessary for me. It was a very pleasant way of working together and I hope that there will be future projects that help us to stay in contact. I would also like to thank Univ.-Prof. Dr.-Ing. Peter Jax for being the second examiner and Prof. Dr.-Ing. Dipl.-Wirt. Ing. Stephan Pachnicke and Prof. Dr.-Ing. Peter Höher for being part of the examination board.

Another huge thank you goes to Dr.-Ing. Markus Buck and Dr.-Ing. Friedrich Faubel, who supported me through all my time at Nuance and later Cerence. I could always ask for help or for a discussion on all related topics and it was a great atmosphere of working together.

Furthermore, I want to thank Dipl.-Ing. Simon Graf and all the colleagues at Nuance and later Cerence for the helpful feedback and the valuable discussions; Dr.-Ing. Tim Haulick for making this dissertation possible in the area of acoustic speech enhancement at Nuance Communications, later Cerence; M. Sc. Tobias Huebschen and all the colleagues at the Institute for Digital Signal Processing and System Theory at Kiel University for the really warm welcome and the interesting conversations; and my wife Solveig Jungclaussen for her constant support, especially in the time when I had to finish this thesis besides working at full scale.

Abstract

Although mobile wideband telephony has been standardized for over 15 years, many countries still do not have a nationwide network with good coverage. As a result, many cellphone calls are still downgraded to narrowband telephony. The resulting loss of quality can be reduced by artificial bandwidth extension. There has been great progress in bandwidth extension in recent years due to the use of neural networks. The topic of this thesis is the enhancement of artificial bandwidth extension using neural networks. A special focus is given to hands-free calls in a car, where the risk is high that the wideband connection is lost due to the fast movement.

The bandwidth of narrowband transmission is not only reduced towards higher frequencies above 3.5 kHz but also towards lower frequencies below 300 Hz. There are already methods that estimate the low-frequency components quite well, which will therefore not be covered in this thesis.

In most bandwidth extension algorithms, the narrowband signal is initially separated into a spectral envelope and an excitation signal. Both parts are then extended separately in order to finally combine both parts again. While the extension of the excitation can be implemented using simple methods without reducing the speech quality compared to wideband speech, the estimation of the spectral envelope for frequencies above 3.5 kHz is not yet solved satisfyingly. Current bandwidth extension algorithms are just able to reduce the quality loss due to narrowband transmission by a maximum of 50% in most evaluations.

In this work, a modification for an existing method for excitation extension is proposed which achieves slight improvements while not generating additional computational complexity. In order to enhance the wideband envelope estimation with neural networks, two modifications of the training process are proposed. On the one hand, the loss function is extended with a discriminative part to address the different characteristics of phoneme classes. On the other hand, by using a GAN (generative adversarial network) for the training phase, a second network is added temporarily to evaluate the quality of the estimation.

The neural networks that were trained are compared in subjective and objective evaluations. A final listening test addressed the scenario of a hands-free call in a car, which was simulated acoustically. The quality loss caused by the missing high frequency components could be reduced by 60% with the proposed approach.

Kurzfassung

Obwohl die mobile Breitbandtelefonie bereits seit über 15 Jahren standardisiert ist, gibt es oftmals noch kein flächendeckendes Netz mit einer guten Abdeckung. Das führt dazu, dass weiterhin viele Mobilfunkgespräche auf Schmalbandtelefonie heruntergestuft werden. Der damit einhergehende Qualitätsverlust kann mit künstlicher Bandbreitenerweiterung reduziert werden. Das Thema dieser Arbeit sind Methoden zur weiteren Verbesserungen der Qualität des erweiterten Sprachsignals mithilfe neuronaler Netze. Ein besonderer Fokus liegt auf der Freisprech-Telefonie im Auto, da dabei das Risiko besonders hoch ist, dass durch die schnelle Fortbewegung die Breitbandverbindung verloren geht.

Bei der Schmalbandübertragung fehlen neben den hochfrequenten Anteilen (etwa 3.5–7 kHz) auch tiefe Frequenzen unterhalb von etwa 300 Hz. Diese tieffrequenten Anteile können mit bereits vorhandenen Methoden gut geschätzt werden und sind somit nicht Teil dieser Arbeit.

In vielen Algorithmen zur Bandbreitenerweiterung wird das Schmalbandsignal zu Beginn in eine spektrale Einhüllende und ein Anregungssignal aufgeteilt. Beide Anteile werden dann separat erweitert und schließlich wieder zusammengeführt. Während die Erweiterung der Anregung nahezu ohne Qualitätsverlust durch einfache Methoden umgesetzt werden kann ist die Schätzung der spektralen Einhüllenden für Frequenzen über 3.5 kHz noch nicht zufriedenstellend gelöst. Mit aktuellen Methoden können im besten Fall nur etwa 50% der durch Schmalbandübertragung reduzierten Qualität zurückgewonnen werden.

Für die Anregungserweiterung wird in dieser Arbeit eine Variation vorgestellt, die leichte Verbesserungen erzielt ohne dabei einen Mehraufwand in der Berechnung zu erzeugen. Für die Schätzung der Einhüllenden des Breitbandsignals mithilfe neuronaler Netze werden zwei Änderungen am Trainingsprozess vorgeschlagen. Einerseits wird die Kostenfunktion um einen diskriminativen Anteil erweitert, der das Netz besser zwischen verschiedenen Phonemen unterscheiden lässt. Andererseits wird als Architektur ein GAN (Generative adversarial network) verwendet, wofür in der Trainingsphase ein zweites Netz verwendet wird, das die Qualität der Schätzung bewertet. Die trainierten neuronale Netze wurden in subjektiven und objektiven Tests verglichen. Ein abschließender Hörtest diente zur Evaluierung des Freisprechens im Auto, welches akustisch simuliert wurde. Der Qualitätsverlust durch Wegfallen der hohen Frequenzanteile konnte dabei mit dem vorgeschlagenen Ansatz um etwa 60% reduziert werden.

Contents

1 Introduction **1**
- 1.1 Motivation and Goals . . . 1
- 1.2 Short Review of the Literature . . . 2
- 1.3 New Contributions in this Work . . . 5
- 1.4 Organization of the Dissertation . . . 6
- 1.5 Notation Conventions . . . 7

2 Basics of Neural Networks **9**
- 2.1 Multilayer Perceptron . . . 10
- 2.2 Training . . . 12
  - 2.2.1 Loss Function . . . 14
  - 2.2.2 Optimization . . . 15
  - 2.2.3 Activation Function . . . 17
  - 2.2.4 Weight Initialization . . . 18
  - 2.2.5 Regularization . . . 19
  - 2.2.6 Time Dependencies in Neural Networks . . . 20
  - 2.2.7 Multi-Condition Training . . . 20
  - 2.2.8 Multi-Task Learning . . . 21
- 2.3 Unsupervised Pre-Training . . . 22
- 2.4 Input Feature Selection Methods . . . 23
- 2.5 Generative Adversarial Networks . . . 24
- 2.6 Hyperparameters . . . 26

3 Speech Production and Transmission **27**
- 3.1 Basics of Speech Production . . . 28
- 3.2 Source-Filter Model of Speech Production . . . 33
- 3.3 Bandwidth of Speech Signals . . . 34
- 3.4 Speech Signal Representations . . . 36
  - 3.4.1 Short-Term Spectrum . . . 36
  - 3.4.2 Mel Spectrum . . . 38
  - 3.4.3 Mel-Frequency Cepstral Coefficients . . . 38
  - 3.4.4 Codecs for Speech Transmission . . . 39

4 Model-Based Artificial Bandwidth Extension System **41**
4.1 Separation of Envelope and Excitation 42
4.2 Extension of the Excitation 44
4.3 Extension of the Spectral Envelope 46
4.4 Synthesis of the Extended Signal 49

5 Enhanced Extension of the Excitation Signal **51**
5.1 Multiple Spectral Shifting 53
5.2 MSS with Comfort Noise 55

6 Improved DNN Training for Spectral Envelope Extension **57**
6.1 Data Generation 58
6.2 DNN Feature Selection 59
6.2.1 Target Feature Selection 60
6.2.2 Input Features 61
6.2.3 Comparison of TD and FD Features 65
6.2.4 Forward Selection for ABE 67
6.3 DNN Training Setup 68
6.4 Discriminative Training 69
6.5 Adversarial Training 71
6.5.1 GAN Training 72
6.5.2 CGAN Training 73
6.5.3 Discriminative CGAN Training 73

7 Evaluation **75**
7.1 Evaluation Setup 75
7.1.1 Artificial Bandwidth Extension Setup 76
7.1.2 Training Data Generation 76
7.1.3 Training Process 78
7.2 Evaluation Metrics 79
7.2.1 Subjective Metrics 80
7.2.2 Objective Metrics 81
7.3 Input Feature Selection 84
7.3.1 Training Setup 84
7.3.2 Results 85
7.4 Excitation Extension Methods 87
7.4.1 Subjective Listening Test 87
7.4.2 Discussion 90
7.5 Discriminative Training 90
7.5.1 Objective Quality Measures 91
7.5.2 Subjective Listening Tests 92
7.5.3 Conclusion 94
7.6 Combination with Adversarial Training 94

7.6.1 Objective Quality Measures . . . 94
7.6.2 Subjective Listening Tests . . . 96
7.6.3 Discussion . . . 98
7.7 ABE in Simulated Driving Situation . . . 98
7.7.1 Subjective Listening Tests . . . 99
7.7.2 Discussion . . . 102

8 Conclusion and Outlook **103**
8.1 Conclusion . . . 103
8.2 Outlook . . . 104

A Abbreviations and Notation **107**
List of Abbreviations . . . 107
Notation . . . 110
List of Latin Symbols . . . 111
List of Greek Symbols . . . 114
Indices . . . 115

List of Figures **117**

List of Tables **119**

Bibliography **121**

Chapter 1

Introduction

The quality and performance of mobile internet has developed rapidly over the last years. But at the same time, many mobile telephony systems are still based on ancient cellular networks. These only support narrowband (NB) connections with a sample rate of 8 kHz, while modern wideband (WB) speech is transmitted at a sample rate of 16 kHz. According to the Nyquist-Shannon sampling theorem, the maximum possible acoustical bandwidth[1] is 4 kHz for NB calls and 8 kHz for WB calls. The bandwidth is further reduced by anti-aliasing filters that stem from the time of analog transmission. In the worst case, a bandpass with cutoff frequencies at 300 Hz and 3.4 kHz is applied to the NB signal by the cellphone in sending direction. Compared to WB speech (about 50 Hz up to 7 kHz), the bandwidth limitation leads to a decreased speech quality and to a 'muffling' sound. Especially today, where a lot of possibilities exist for high-quality phone calls or even video calls over mobile internet, the NB speech quality is not acceptable. In order to support WB transmission, all older hardware had to be updated or replaced, which is a rather slow process. And a WB call can only be set up when all involved hardware parts support WB calls. As a consequence, many calls are still transmitted through NB channels although most of the end-user devices support higher sample rates. The probability that no WB transmission is available rises even more when a person is moving, especially in rural areas with a weak network coverage. This can also lead to a fall back from WB to NB transmission while a call is running.

1.1 Motivation and Goals

The drop of speech quality that is caused by a NB connection can be reduced by artificial bandwidth extension (ABE), an algorithm that aims at reproducing the missing parts of the speech spectrum after transmission. Therein, a NB signal is converted to a WB signal by estimating and adding the spectral content that was not transmitted. ABE is the only way to recover the WB quality of a NB call without changing the network equipment and the sending hardware. Although ABE has been a research topic for

[1] In the following, the term *bandwidth* will always refer to *acoustical bandwidth*.

decades now, the signal quality is on average still not close to real WB quality. This can be caused by disturbing artifacts that are introduced when the extension does not work accurately and by predicting less energy than it would be necessary in some frames.

ABE and speech processing in general have been active research topics for a long time. In the last years, more and more of these solutions have been improved by using neural networks. This trend is aligned with the growing success of neural networks in general. The high computational power of modern graphical processors allows for a fast training of complex deep neural networks (DNNs). The main goal of this thesis is to further enhance these approaches in terms of speech quality and robustness.

In this work, a special focus is put on phone calls in a car, where people are moving fast and therefore have a high risk that the phone call quality is poor. This means that the algorithm has to run efficiently with minimal requirements regarding processing power and memory. Recent publications show a trend towards big networks with many wide layers which stands in contrast to these limitations. This thesis tries to find a good compromise between optimal performance and small network dimensions.

DNNs have the drawback that the internal variables are hard to interpret and that it is difficult to exactly understand what will be predicted by the network in which situation. This becomes a major problem when situations emerge in the everyday use in which the algorithm does not work as expected. Therefore, a robust performance is another goal of this thesis.

It is important to note that this thesis does not claim to be comprehensive in showing up possible approaches of ABE using neural networks. All decisions were taken while having in mind that the final solution should at the same time run in real-time on small processors and yield a good speech quality.

1.2 Short Review of the Literature

The early approaches of extending the bandwidth of a speech signal were based on basic signal-processing methods. Maybe the first contribution to an artificial extension of the speech bandwidth was made by Schmidt in 1933 [Sch33]. An application for ABE was published first by the BBC in 1972 [Cro72]. In 1979, Makhoul et al. presented methods to regenerate the missing upper band (UB) excitation above 4 kHz by spectral duplication [MB79]. These methods are still often used to extend the excitation signal. In 1983, another signal-processing based ABE approach was proposed by Patrick [Pat83]. However, none of these early approaches yielded a satisfying speech quality.

In 1992 and 1994, new approaches were published [COM92; CH94; COM94; YA94] that included the decomposition of the speech signal in its envelope and its excitation based on the source-filter model of speech production [Fan60]. This decomposition has been applied in many approaches and is still widely used. The ABE algorithms presented in this thesis also rely on this basic principle. In 1995, the extension of the NB signal was also applied towards lower frequencies [AHW95]. Avendano et al. estimated the cepstral coefficients, which were based on linear predictive coding (LPC),

of an all-pole system that modeled the WB spectral envelope with a finite impulse response (FIR) filter applied to the NB cepstral coefficients. This method was used for the recovery of spectral contents in low and high frequencies. The excitation was extended using spectral folding (SF). Codebook mapping based on line spectral pairs (LSPs) was used to reproduce the higher frequencies of a narrowband code excited linear prediction (CELP)-coded speech signal in [CH96]. In 1997, Nakatoh et al. compared ABE methods that estimate the UB using piecewise linear mapping, codebook mapping, and neural networks [NTN97]. As the size of neural networks was very restricted at those times ($15 \times 45 \times 45 \times 15$ neurons in [NTN97]), they did not yield better results than the competing methods. Subjective listening tests proved that the bandwidth was perceived to be broader by most listeners, but the speech quality of NB was still preferred. Epps and Holmes achieved further improvements by codebook mapping in 1999 [EH99]. The best results were obtained when the codebooks were split into two sets of codebook vectors for voiced and unvoiced speech frames. In the same year, another codebook mapping approach was presented, which was based on mel frequency cepstral coefficients (MFCCs) instead of LPCs [EK99]. In a subjective test, 86% of the listeners preferred the overall speech quality of the reconstructed WB to NB [EK99]. Later approaches used Gaussian mixture models (GMMs) to model the spectral envelope, in combination with a joint density estimation technique [PK00; QK03] or with a hidden Markov model (HMM) that selects codebook entries [JV00; JV03a]. A general overview of the contributions in the field of ABE up to 2002 was given in the Ph.D.-thesis by Jax [Jax02]. In the current work, the input features for ABE systems that were proposed in [Jax02; JV04] are taken as base for an input feature selection approach for neural networks. In 2003, an ABE system based on spectral envelopes, estimated by neural networks, was able to improve the quality of the NB signal subjectively [IS03]. However, in a direct comparison, a codebook approach with a higher complexity still yielded better results. A phoneme-dependent codebook mapping of line spectral frequencies (LSFs) was implemented in [HKA05]. In [VZY06], the harmonicity in the UB is estimated jointly with parameters for the excitation and the envelope in a codebook approach in order to enhance the excitation extension. A different technique, in which an UB spectrum is first created using SF and then modified using parameters that are predicted by a neural network, was proposed in [KLA07]. An extensive comparison of different approaches for envelope and excitation extension was made in 2008 [IMS08].

From 2010 on, neural networks have dominated the field of ABE more and more. An approach from the year 2011 successfully predicted four mel-frequency bands that describe the UB spectral envelope with a neural network [PA11]. An important step regarding the objective evaluation of ABE methods was made by Möller et al. [Möl+13]. The subjective evaluation of various ABE methods showed that none of the three objective measures WB-PESQ (WB perceptual evaluation of speech quality), POLQA (perceptual objective listening quality analysis), and DIAL (diagnostic instrumental assessment of listening quality) were able to reliably predict the subjective rank order of different ABE systems. These results were supported by [Bau+14; Pul+15] regarding

WB-PESQ and POLQA. In 2014, Bauer et al. proposed to use two neural networks in a HMM approach for ABE [BAF14]. One of the networks had the task to detect the phoneme class consisting of [s] and [z][2] in order to introduce enough energy in the UB for those frames. Li et al. showed that feedforward DNNs outperform GMMs for ABE [LL15]. Gu and Ling came to the same result and also found that DNNs yield better results than restricted Boltzmann machines (RBMs) and bidirectional associative memories (BAMs) [GL15]. In extensive listening tests, Abel et al. compared six ABE methods and reported language-specific results for the four selected languages English, German, Chinese, and Korean [Abe+16]. The quality gap between adaptive multi rate (AMR)-NB (at 12.2 kbps) and AMR-WB (at 23.05 kbps) could be closed by the best approaches in English, German, and Korean by 46%, 25%, and 36%, respectively [Abe+16]. Probably the first contributions that used recurrent neural networks (RNNs) to model the time dependency between adjacent frames while estimating the UB spectral envelope were presented in 2017 [Wan+16; GLD16]. While Wang et al. modeled the time dependencies with a recurrent temporal restricted Boltzmann machine (RTRBM) in combination with a DNN, Gu et al. used long-short term memory (LSTM) layers. The recurrent model in [GLD16] could further be enhanced by applying a bottleneck layer in a DNN that yielded good input features for the LSTM layer.

A first objective measure for ABE that correlates well with subjective scores was proposed in 2017 by Abel et al. [Abe+17]. They trained a DNN on different ABE methods to predict the subjective ratings that were originally gained through listening tests. In the same year, stacked dilated convolutional neural networks (CNNs) were applied to ABE by Gu and Ling [GL17]. The difference compared to most prior approaches is that the processing was done on the waveform directly instead of processing the signal frame by frame. Another approach that does not use a short-term Fourier transform (STFT) is the ABE method by Bachhav et al., which is based on the constant Q transform (CQT) [Bac+17]. Regarding frame-based DNNs, a regression network gave better results than an HMM/GMM-based classification network [AF17].

In 2018, an enhanced excitation extension method was proposed in a work related to this thesis [Sau+18a]. Later that year, the results of a feature selection approach for ABE using neural networks was presented in another related contribution [SFS18]. A third publication focused on the discriminative training of a DNN for ABE, which is, like the two preceding approaches, explained in detail in this thesis [Sau+18b]. Lee et al. proposed an ABE system that does not separate the NB signal into spectral envelope and excitation but duplicates the NB part by SF and then modifies the energy in every subband [Lee+18]. Therein, an ensemble of sequential networks was trained to apply speech enhancement on the NB input and to subsequently estimate the subband energy. The output of a separate DNN, which was used to predict voiced frames, was used to select a sequential network, trained on only voiced or only unvoiced speech sounds. An approach that focuses on low delay and low complexity was presented by Schmidt and

[2]The notation of phonemes in this thesis is chosen according to the international phonetic alphabet (IPA) [Int99].

Edler [SE18]. They used a combination of a CNN and LSTM cells to model both, the dependencies between adjacent frames and adjacent frequency bins, accurately. Li et al. were the first to use generative adversarial networks (GANs) for the prediction of the WB spectral envelope in ABE [Li+18]. This approach was further enhanced by the proposed conditional GAN (CGAN) with discriminative training in [Sau+19]. This method is also part of this thesis.

1.3 New Contributions in this Work

The author's contributions in this thesis and the related papers that have been published are listed in the following paragraphs. Note that these contributions were published under the author's name of birth, Jonas Sautter, instead of his current name, Jonas Jungclaussen.

Enhanced Excitation Extension Methods The same methods for excitation extension that have been used for several decades are still often used in recent publications. These methods are called spectral folding (SF) and spectral shifting (SS). In a prior work [Sau+18a], two variations of the classical method of SS were proposed, namely multiple spectral shifting (MSS) and multiple spectral shifting with comfort noise (MSSCN). These methods are thought to reduce artifacts that are inserted by SS while performing ABE. A detailed description of these methods is given in chapter 5. Subjective listening tests prove the small benefit in terms of speech quality. They also show that no further investigations seem to be necessary, as the quality of MSS cannot reliably be distinguished from original WB speech quality. The quality gap between recent ABE methods and real WB is much larger in terms of the extension of the spectral envelope, which therefore leaves more space for improvements.

Input Feature Selection for DNN-Based Envelope Extension Many recent publications describe DNN trainings in which all input data is fed to the network in its raw and unprocessed form. If complex features are necessary for a given task, a DNN is generally able to learn some kind of feature extraction in its first layer automatically. But as the goal of this thesis and the related work is to develop an efficient algorithm, the high cost for the extra layer in the DNN was avoided by applying an efficient feature extraction. The main drawback is that the network cannot choose the best features for the task on its own any more. The input features were chosen based on a feature selection method from a pool of features that was created beforehand. DNNs were trained with many feature combinations and the performance after training was evaluated. The contribution lies in the application of the given feature selection method to the problem of ABE and in defining some frequency domain (FD) speech features that are especially robust against background noise. The feature selection approach for DNN-based ABE was first presented in [SFS18].

Discriminative Training for DNN-Based Envelope Extension A regression DNN has been used in most recent ABE approaches to estimate the energy distribution in the UB. An extension for the loss function was proposed in [Sau+18b] in order to better represent the subjective perception of the speech signal quality. This extension forces the network to distinguish between different phoneme classes. In the related paper, especially the characteristics of the phoneme class of sibilant fricatives was investigated, because the UB energy is much higher for sibilant fricatives than for other phonemes. If this energy is underestimated, a lisping sound will be generated, which manifests in a strong degradation of the subjective quality. Subjective and objective evaluations proved the advantages of the proposed method.

Adversarial Training for DNN-Based Envelope Extension The problem that the regression DNN did not distinguish enough between different phonemes could partly be solved by discriminative training. However, the results still showed a tendency that the inserted energy did not vary enough over time. In order to estimate more realistic energy distributions for the UB, adversarial training was applied to ABE [Sau+19]. This was the first application of a CGAN to ABE and it was the second time that a GAN was used for ABE in general. The training was furthermore successfully combined with discriminative training. The effectiveness of the combined approach was shown by objective metrics and in a subjective listening test.

1.4 Organization of the Dissertation

A basic overview of neural networks that is necessary for a good understanding of this thesis is given in chapter 2. The architecture of a simple multilayer perceptron (MLP) and the principle of its training process are outlined exemplarily. Some DNN variants and enhanced training methods are introduced.

Chapter 3 covers the theoretical background for the generation and transmission of human speech. After comprising some basics from the field of linguistics on distinguishing different phonemes and the respective speech sounds, the process of speech production is formulated as source-filter model. In the field of speech transmission, some representations of speech signals that are used for speech transmission are explained after having introduced the characteristics of the bandwidth of transmitted speech.

Chapter 4 provides the description of a general system for model-based ABE. This includes the tasks of separating the NB signal into an excitation signal and a spectral envelope, extending both separately, and finally synthesizing the predicted WB signal out of the NB signal and both extended parts.

In chapter 5, some classical excitation extension methods are introduced and a novel method is proposed. All methods presented in this chapter are based on spectrally shifting or mirroring parts of the spectrum. The problems that still exist when using these classical methods are discussed and partly alleviated in the proposed method. An insertion of white noise towards higher frequencies is additionally proposed in order to

model the WB excitation more accurately. The results of these methods are presented in chapter 7.

In chapter 6, improved training methods for the prediction of an extended spectral envelope are proposed. In a first part, the generation of the training data is described. A second part deals with the selection of good features for an ABE regression network and the DNN training setup. The third part contains two novel modifications. In the first modification, a discriminative training approach is explained that focuses on the correct energy distribution for different phoneme classes. The second approach applies adversarial training to ABE in order to achieve more realistic predictions for the WB spectral envelope.

In chapter 7, the evaluation of the proposed methods from chapters 5 and 6 is presented. At first, the general evaluation setup is described, which is used in all of the following evaluations. In a second step, all metrics that are necessary are defined. Objective and subjective criteria are used in order to get an unbiased comparison. In a final listening test, the combination of all proposed approaches is evaluated in a simulated acoustical environment of driving in a car.

A conclusion and an outlook with possibilities for further improvements are given in chapter 8.

1.5 Notation Conventions

In this thesis, upper-scale bold letters like $\mathbf{X}$ denote matrices, lower-scale bold letters like $\mathbf{x}$ denote vectors, and letters in italics like x denote scalars. There is a short list of exceptions of scalar variables that have a variable name in upper-scale:

- lengths of vectors or numbers of elements,
- short-term spectra (S, Z),
- loss functions (J), and
- neural network identifiers (C, D, D_c, G, G_c, R, R_d).

A single element in a matrix is normally denoted by its indices in subscript like in x_{i_1,i_2}. An exception is made for the time and the frequency dimension, where an element is written as a signal with the frequency index k and the time index l, like in $S(k,l)$. The apostrophe $(\cdot)'$ denotes that the respective variable refers to the mel scale instead of the linear frequency scale. All variants of i refer to integer index variables. Single characters surrounded by squared brackets like [s] refer to the phonemes of the phonetic alphabet IPA [Int99].

For most signals, different variants exist. These are mostly specified as subscript or superscript indices. A list with all indices as well as a general explanation of the notation and the used symbols is given together with the list of abbreviations in appendix A.

Chapter 2

Basics of Neural Networks

Today, artificial neural networks (ANNs) dominate the research topics in many areas of signal processing. The reason for this is that they are able to mimic non-linear and very complex relations. This comes to the cost of a high number of parameters that have to be calculated in an iterative process. With the rapidly rising computational power of modern graphics processors, this requirement is fulfilled more and more. Besides the computational power, a high amount of representative data is necessary for the training of an ANN. This might be the limiting factor in many of today's machine learning (ML) approaches. A large set of input values, often combined with the respective optimal output values, is required for learning complex relations. The optimal output values are called targets in the scope of this thesis.

A common way of using ANNs in an application is to train the network in a first stage and to apply the trained network to the application in a second stage. These two stages are called training stage and prediction stage. In other words, the parameters of the network do not change any more once the prediction stage is reached. Other approaches continue to update the parameters online in order to adapt to specific situations. In this work, online learning methods will not be covered because all networks are first trained and then integrated in an application.

In general, there are two main tasks that can be solved with ANNs: classification and regression tasks. In a classification task, samples of input data are assigned to discrete target classes. In contrast to this, regression tasks predict continuous target values based on the input data. There are many specific variants of ANNs today and many different architectures and types of layers have been developed. Only the few network types that are used in this work are described in this chapter. A basic ANN, the multilayer perceptron (MLP), is explained briefly in section 2.1 in order to explain the fundamental principles of ANNs.

Machine learning can also be categorized depending on the availability of labeled data. A network training where the target data is known completely belongs to the class of supervised learning. Unsupervised and semi-supervised learning is applied when the targets are unknown or just a small subset of the data is labeled with target data. In the scope of this thesis, the focus lies on supervised learning because all speech

samples can be used as target data and because the input data can be created from target data easily. Consequently, the other categories will not be addressed.

Training ANNs is an iterative process. In a first step, initial values have to be chosen for all network variables. In every training iteration, the ANN predicts output values based on the given input values and the internal variables. This prediction is compared with the respective target values and the prediction error is evaluated with a loss function. The result of this function shall be minimized by adjusting the variables in the network. The whole training process is described in more detail in section 2.2.

Depending on the training setup, the initial values of the network can have a high influence on the convergence [GB10]. In some settings, the convergence can be substantially improved by pre-training the network variables [Erh+10]. Unsupervised pre-training using stacked auto-encoders is described in section 2.3.

It is a complex task to define a loss function that reflects the perceptual distance between prediction and target values. A standard loss function for regression tasks is the mean square error (MSE). Pure MSE-based training has some drawbacks that will be discussed in chapter 6. GANs were trained in this thesis to overcome these drawbacks. The training of a GAN is addressed in section 2.5.

A lot of parameters can be controlled when training an ANN. In most cases, there is no rule which determines the optimal values for a given problem. Section 2.2 deals with methods to find reasonably good values for hyperparameters of the network.

2.1 Multilayer Perceptron

Most of today's common network architectures are based on the multilayer perceptron (MLP) which was already proposed in 1962 [Ros62]. The MLP is a network that maps a vector of input values[1] $\mathbf{x} = [x_0, \ldots, x_{N_\mathbf{x}-1}]^\mathsf{T}$ to an output vector $\hat{\mathbf{y}} = \left[\hat{y}_0, \ldots, \hat{y}_{N_\mathbf{y}-1}\right]^\mathsf{T}$. The network consists of artificial neurons that are organized in $\mathcal{L}$ layers. Figure 2.1 depicts the structure of an MLP based on its neurons and the connections between them. The neurons are named by their respective output activations $a_{i^{[\lambda]}}^{[\lambda]}$, where $i^{[\lambda]}$ is the neuron index and $\lambda \in \{0, \ldots, \mathcal{L}\}$ is the layer index. The vector of activations at the output of a layer λ is then

$$\mathbf{a}^{[\lambda]} = \left[a_0^{[\lambda]}, \ldots, a_{N^{[\lambda]}-1}^{[\lambda]}\right]^\mathsf{T}. \tag{2.1}$$

The term activation denotes the value that is sent from one neuron to another in the scope of this thesis. The vector of activations in the input layer is equal to the input data vector and the vector of activations in the last layer is equal to the predicted output vector:

$$\mathbf{a}^{[0]} = \mathbf{x} \tag{2.2}$$

$$\mathbf{a}^{[\mathcal{L}]} = \hat{\mathbf{y}} \tag{2.3}$$

[1] Note that, in this section, all formulas will be independent of the sample index l and the network parameters Θ for a better readability.

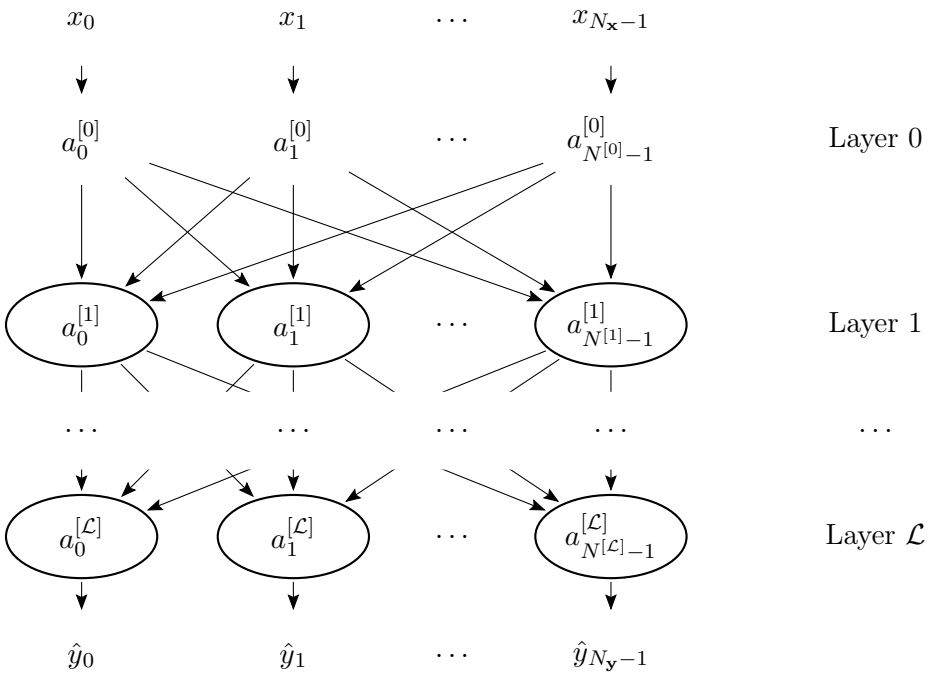


Figure 2.1: MLP graph with $\mathcal{L}$-1 hidden layers. The artificial neurons are named by their respective output activations $a_{i^{[\lambda]}}^{[\lambda]}$ for layer λ and index $i^{[\lambda]}$. The input and output vectors are equal to the respective activations in the first and last layer: $\mathbf{x} = \mathbf{a}^{[0]}$ and $\hat{\mathbf{y}} = \mathbf{a}^{[\mathcal{L}]}$.

Accordingly, the number of coefficients has to match the lengths of the data vectors: $N^{[0]} = N_\mathbf{x}$ and $N^{[\mathcal{L}]} = N_\mathbf{y}$. The input layer ($\lambda = 0$) just forwards the input activations $\mathbf{x}$ to the next layer ($\lambda = 1$). Each of the $N^{[\lambda]}$ neurons in layer λ with $\lambda > 0$ is connected to all $N^{[\lambda-1]}$ neurons of the previous layer.

A detailed structure of an artificial neuron is shown in fig. 2.2. For the neuron at index $i^{[\lambda]}$ in layer λ, the sum of all incoming activations and an optional bias term $b_{i^{[\lambda]}}^{[\lambda]}$ is

$$v_{i^{[\lambda]}}^{[\lambda]} = \sum_{i^{[\lambda-1]}=0}^{N^{[\lambda-1]}-1} w_{i^{[\lambda]},i^{[\lambda-1]}}^{[\lambda]} \cdot a_{i^{[\lambda-1]}}^{[\lambda-1]} + b_{i^{[\lambda]}}^{[\lambda]}, \tag{2.4}$$

where the activation coming from neuron $i^{[\lambda-1]}$ in layer $\lambda-1$ is the product of the output activation $a_{i^{[\lambda-1]}}^{[\lambda-1]}$ and a weighting factor $w_{i^{[\lambda]},i^{[\lambda-1]}}^{[\lambda]}$. In vector notation, the weights at layer λ can be written as

$$\mathbf{W}^{[\lambda]} = \begin{bmatrix} w_{0,0}^{[\lambda]} & w_{0,1}^{[\lambda]} & \cdots & w_{0,N^{[\lambda-1]}-1}^{[\lambda]} \\ w_{1,0}^{[\lambda]} & w_{1,1}^{[\lambda]} & \cdots & w_{1,N^{[\lambda-1]}-1}^{[\lambda]} \\ \vdots & \vdots & \ddots & \vdots \\ w_{N^{[\lambda]}-1,0}^{[\lambda]} & w_{N^{[\lambda]}-1,1}^{[\lambda]} & \cdots & w_{N^{[\lambda]}-1,N^{[\lambda-1]}-1}^{[\lambda]} \end{bmatrix}. \tag{2.5}$$

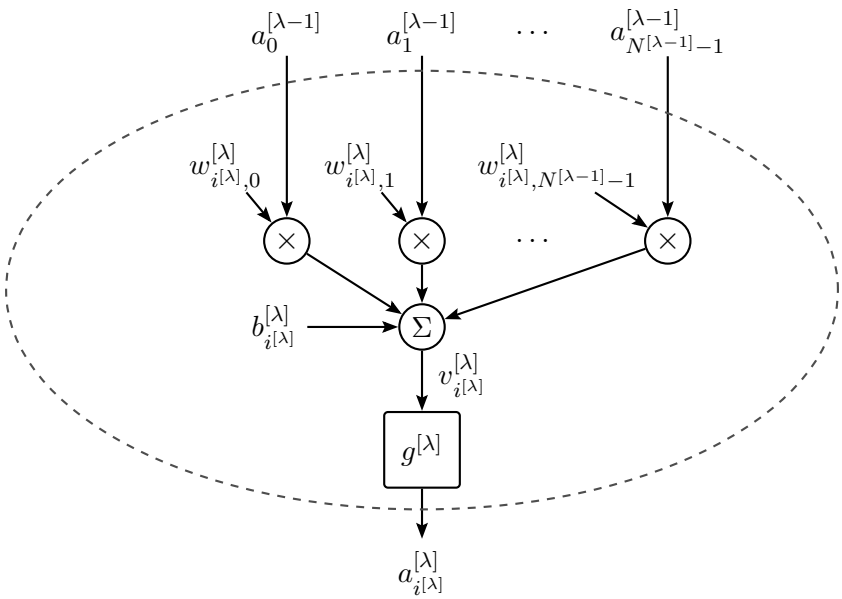


Figure 2.2: Artificial neuron at index $i^{[\lambda]}$ in layer λ of an MLP for $\lambda > 0$. In the input layer ($\lambda = 0$), the inputs are just forwarded to the next layer. Note that there are exceptions in which the activation function $g^{[\lambda]}$ depends on the whole vector $\mathbf{v}^{[\lambda]}$.

Equation (2.4) can then be simplified to

$$\mathbf{v}^{[\lambda]} = \mathbf{W}^{[\lambda]}\mathbf{a}^{[\lambda-1]} + \mathbf{b}^{[\lambda]}. \tag{2.6}$$

An activation function $g^{[\lambda]}$, which is non-linear in most cases, is evaluated on this sum in order to compute the output activation

$$a_{i^{[\lambda]}}^{[\lambda]} = g^{[\lambda]}(v_{i^{[\lambda]}}^{[\lambda]}). \tag{2.7}$$

The activation function is necessary for the MLP to be able to learn non-linear dependencies. Exceptions exist, in which the activation function $g^{[\lambda]}$ depends on the whole vector $\mathbf{v}^{[\lambda]}$ and cannot be written element-wise. The internal variables of an MLP are described as

$$\Theta = \{\mathbf{W}^{[\lambda]}, \mathbf{b}^{[\lambda]}, g^{[\lambda]}\} \qquad \forall\lambda \in \{1, \ldots, \mathcal{L}\} \tag{2.8}$$

in the following, the type of the activation function is also interpreted as parameter. MLPs can have multiple layers that all have the form described above. Because of its structure, an MLP layer is also called fully-connected feedforward layer. As there is no precise definition for the term DNN, it will be used in this thesis when referring to networks with at least one hidden layer. The layer types and the connections of DNNs are not restricted to a special form in contrast to the layers and connections of an MLP.

2.2 Training

The training of a DNN aims at finding suitable values for the variables Θ in an iterative fashion. In supervised learning, a tuple of input and target data is used in every

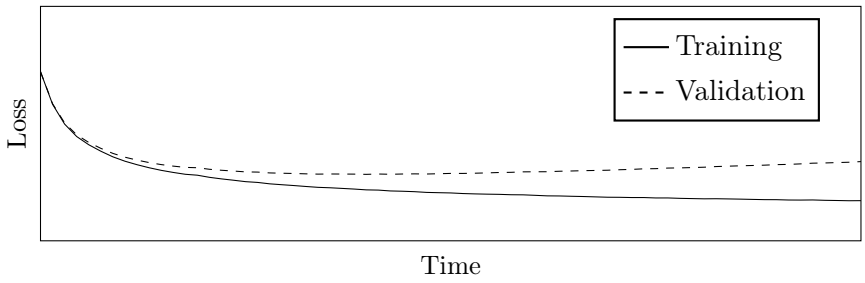


Figure 2.3: The training loss (solid line) and the validation loss (dashed line) are plotted for the duration of a training process. It can be seen that the validation error stops to decrease at a certain point and increases again. This is a clear signal for overfitting in the training process.

iteration. The input data $\mathbf{x}$ is fed into the network and the activations are propagated to the output layer following the rules explained above. The predicted values $\hat{\mathbf{y}}(\Theta)$ are then compared to the target values $\mathbf{y}$ and a loss function $J(\Theta)$ is evaluated. Standard loss functions and their role in the training process are treated in section 2.2.1.

Finally, all weights and biases in the DNN are updated in such a way, that the loss is minimized. The basic principles of updating the weights following this optimization criterion are explained in section 2.2.2.

Activation functions enable DNNs to learn non-linear dependencies. A large variety of activation functions exists, of which a small subset of relevant functions for this work is presented in section 2.2.3.

One propagation through all training samples is called one epoch. A stopping criterion can be defined based on the convergence. Before the iterative process of updating the weights starts, all variables of the network have to be initialized. The initialization can have a major influence on the training results. Some common methods for the initialization of the weights are outlined in section 2.2.4.

A DNN is able to learn exact mappings of input and output data in case the complexity of the network is high compared to the amount of training data. This effect is called overfitting. It is undesirable as the principle behind the training data shall be learned instead of the data itself. Regularization methods have the aim to reduce this effect, they are introduced briefly in section 2.2.5.

The amount of overfitting can be observed by evaluating the error of the DNN on another dataset, called validation set (see fig. 2.3). The results of this evaluation can further be used to control the convergence process. In order to keep a part of the data that is not used in the training process, a small additional test set can be defined. The training set is normally the main part of the data, validation and test set are often set to around 10–20% of the whole dataset. When the training input data does not cover all variations of real input data (which is often the case), it is hard to predict what happens when a new variation of the data occurs. Means for achieving a high variety regarding various properties of speech data for ABE are presented in section 2.2.7.

A DNN is often compared to a black-box whose internal logic can never be fully

understood. It happens frequently that a network focuses on a certain detail of the input data that is just by coincidence well correlated to the target data in the training dataset. If this detail is not generally correlated to its target, the training dataset is not representative and should be expanded. However, finding all these details is a complex task and the training dataset might never cover all situations that occur in the prediction stage. Another way to reduce the unwanted effect of convergence to a local minimum is to check whether the same network would perform well on different but related tasks. In multi-task learning (MTL), the network learns to predict multiple targets at the same time, see section 2.2.8. It was found to be beneficial in many kinds of DNNs [Car97] and regarding speech processing especially in the field of speech recognition since 2003 [PG03].

2.2.1 Loss Function

A loss function $J(\Theta)$ defines an error measure between a target feature vector $\mathbf{y}$ and a predicted feature vector $\hat{\mathbf{y}}(\Theta)$. This error shall be minimized in the training process by adjusting the weights in the network appropriately. Depending on the type of optimization, the loss function is either calculated on one training sample or on a batch of samples. This batch can again be subdivided into mini-batches that consist of M samples. A general form of the loss function can be defined dependent on a sample index $l_m \in \{0, \ldots, L-1\}$ and a mini-batch index $m \in \{0, \ldots, M-1\}$. Because the input and the output data of the DNN for ABE are time series, the time frame index $l = mL + l_m$ will be used in the following. This makes it possible to formulate functions in dependency of l or m.

Most algorithms of speech signal enhancement are optimized to yield high subjective ratings by human listeners. Consequently, a loss function of a DNN that shall enhance a speech signal should represent especially those measures that highly correlate with subjective ratings. Finding such a loss function is one of the most important tasks when training a DNN.

In ABE, the DNN predicts the energy distribution of the UB (see section 1.2). In most recent publications, a regression DNN predicts this distribution, earlier approaches selected a codebook entry in a classification task (see section 1.2). Two loss functions that have proven its worth are the cross-entropy for classification tasks and the mean square error (MSE) for regression tasks. For both functions, the loss $\bar{J}(m, \Theta)$ for a mini-batch m is the average loss over all L frames:

$$\bar{J}(m, \Theta) = \frac{1}{L} \sum_{l=mL}^{(m+1)L-1} J(l, \Theta) \tag{2.9}$$

Mean Square Error Training The MSE is chosen as loss function for many non-specific regression tasks. Higher deviations in the prediction have a high influence on the total loss because of the squared characteristics. The MSE at frame index l can be

written as[2]

$$J^{\text{mse}}(l,\Theta) = \|\mathbf{y}(l) - \hat{\mathbf{y}}(l,\Theta)\|_2^2 \tag{2.10}$$

$$= \sum_{i=0}^{N_{\mathbf{y}}-1} |y_i(l) - \hat{y}_i(l,\Theta)|^2. \tag{2.11}$$

Cross-Entropy Training Similarly, the cross-entropy loss function is used for classification tasks. The output layer is mostly a SoftMax layer with one node for each class. All output activations of a SoftMax layer add up to 1 because of the SoftMax activation function, which is explained in section 2.2.3. The target features have to be formatted as one-hot encoded vectors, where all values except for the target class are zero while the value for the target class is one. The cross-entropy loss function at frame index l is defined as

$$J^{\text{ce}}(l,\Theta) = -\sum_{i=0}^{N_{\mathbf{y}}-1} y_i(l) \cdot \log(\hat{y}_i(l,\Theta)). \tag{2.12}$$

2.2.2 Optimization

The loss function returns an error value for each predicted sample. In order to minimize this error, the weights in the network are slightly modified in every iteration. The direction of the modification is contrary to the gradient of the loss function with respect to the respective weight. The size of the update step is defined by the size of the gradient multiplied with the learning rate or step size η. This general training procedure is also known as gradient descent (GD) [RHW86a]. An adaptation of η throughout the training process can improve the convergence speed and the final result [KB14].

The gradients in a DNN are usually calculated by the well known backpropagation algorithm [RHW86b], which is exemplarily done in the following for a feedforward regression DNN. The aim of the algorithm is to calculate the partial derivative of the loss with respect to all weights and biases in the network. The frame index l is constant for the following equations and will therefore be omitted. The same holds for the network parameters Θ. Starting with the last layer, the partial derivative of the loss function with respect to the weight $w^{[\mathcal{L}]}_{i^{[\mathcal{L}]},i^{[\mathcal{L}-1]}}$ can be calculated with the chain rule of derivation:

$$\frac{\partial J}{\partial w^{[\mathcal{L}]}_{i^{[\mathcal{L}]},i^{[\mathcal{L}-1]}}} = \frac{\partial J}{\partial a^{[\mathcal{L}]}_{i^{[\mathcal{L}]}}} \frac{\partial a^{[\mathcal{L}]}_{i^{[\mathcal{L}]}}}{\partial w^{[\mathcal{L}]}_{i^{[\mathcal{L}]},i^{[\mathcal{L}-1]}}}. \tag{2.13}$$

The first factor describes the partial derivative of the loss function with respect to the predicted output and can easily be calculated. The second factor can be generally

[2]Note that a factor of 1/2 is multiplied to the MSE loss function in some publications. This results in a derivative of $\hat{y}_i(l,\Theta) - y_i(l)$ without a factor of 2.

written for any layer $\lambda > 0$ by again applying the chain rule as

$$\frac{\partial a_{i^{[\lambda]}}^{[\lambda]}}{\partial w_{i^{[\lambda]},i^{[\lambda-1]}}^{[\lambda]}} = \frac{\partial a_{i^{[\lambda]}}^{[\lambda]}}{\partial v_{i^{[\lambda]}}^{[\lambda]}} \frac{\partial v_{i^{[\lambda]}}^{[\lambda]}}{\partial w_{i^{[\lambda]},i^{[\lambda-1]}}^{[\lambda]}}. \tag{2.14}$$

By inserting eq. (2.7), the first factor of eq. (2.14) becomes

$$\frac{\partial a_{i^{[\lambda]}}^{[\lambda]}}{\partial v_{i^{[\lambda]}}^{[\lambda]}} = \frac{\mathrm{d}g^{[\lambda]}(v_{i^{[\lambda]}}^{[\lambda]})}{\mathrm{d}v^{[\lambda]}}. \tag{2.15}$$

It is assumed for the basic description at this point that the activation function g is differentiable although there are activation functions that have non-differentiable points. The second factor in eq. (2.14) is the partial derivative of the weighted sum with respect to the weight $w_{i^{[\lambda]},i^{[\lambda-1]}}^{[\lambda]}$, which is equal to the output at index $i^{[\lambda-1]}$ in the preceding layer:

$$\frac{\partial v_{i^{[\lambda]}}^{[\lambda]}}{\partial w_{i^{[\lambda]},i^{[\lambda-1]}}^{[\lambda]}} = a_{i^{[\lambda-1]}}^{[\lambda-1]}. \tag{2.16}$$

Inserting eqs. (2.15) and (2.16) into eq. (2.14) yields

$$\frac{\partial a_{i^{[\lambda]}}^{[\lambda]}}{\partial w_{i^{[\lambda]},i^{[\lambda-1]}}^{[\lambda]}} = \frac{\mathrm{d}g^{[\lambda]}(v_{i^{[\lambda]}}^{[\lambda]})}{\mathrm{d}v^{[\lambda]}} \cdot a_{i^{[\lambda-1]}}^{[\lambda-1]}. \tag{2.17}$$

The second term vanishes in the calculation of the partial derivative with respect to the bias $b_{i^{[\lambda]}}^{[\lambda]}$, which is an additive term and therefore independent of the input activation:

$$\frac{\partial a_{i^{[\lambda]}}^{[\lambda]}}{\partial b_{i^{[\lambda]}}^{[\lambda]}} = \frac{\mathrm{d}g^{[\lambda]}(v_{i^{[\lambda]}}^{[\lambda]})}{\mathrm{d}v^{[\lambda]}}. \tag{2.18}$$

The partial derivative of a neuron's output activation with respect to the input $a_{i^{[\lambda-1]}}^{[\lambda-1]}$ can be calculated similarly. Substituting $w_{i^{[\lambda]},i^{[\lambda-1]}}^{[\lambda]}$ with $a_{i^{[\lambda-1]}}^{[\lambda-1]}$ in eq. (2.17) yields

$$\frac{\partial a_{i^{[\lambda]}}^{[\lambda]}}{\partial a_{i^{[\lambda-1]}}^{[\lambda-1]}} = \frac{\mathrm{d}g^{[\lambda]}(v_{i^{[\lambda]}}^{[\lambda]})}{\mathrm{d}v^{[\lambda]}} \cdot w_{i^{[\lambda]},i^{[\lambda-1]}}^{[\lambda]}. \tag{2.19}$$

With eqs. (2.17) to (2.19), the gradients for all weights and biases in the last layer as well as its input activations can be calculated. Once this is done, the gradients for the next layer can be calculated through the chain rule and so on. The weights and biases are finally updated with a step of $-\eta$ times the gradient of the loss function:

$$w_{i^{[\lambda]},i^{[\lambda-1]}}^{[\lambda]} \leftarrow w_{i^{[\lambda]},i^{[\lambda-1]}}^{[\lambda]} - \eta \cdot \frac{\partial J}{\partial w_{i^{[\lambda]},i^{[\lambda-1]}}^{[\lambda]}}. \tag{2.20}$$

Usually, problems that are solved using DNNs are not convex. A major drawback of the GD approach is that it leads to a local minimum that can be far away from

the global minimum. A better convergence is achievable by stochastic gradient descent (SGD) [RM51]. In a sample-by-sample SGD, the weights are updated after each training sample. Besides the better convergence, this requires that the gradients are calculated for every sample which consumes a lot of time. Additionally, the convergence is less smooth because the gradient is no more averaged over multiple data points. Mini-batch SGD, which is another variant of SGD, is intended to overcome these disadvantages. The updates are therein applied to a whole mini-batch of M training samples. The gradients can be calculated in parallel and the inverse averaged gradient is finally applied as update step.

In most of the recent papers, a special optimizer called *Adam* [KB14] is implemented for DNN training as modification of the pure mini-batch SGD. The optimizer adaptively controls the learning rate by means of adaptive moment estimation in order to achieve a faster convergence.

2.2.3 Activation Function

An activation function $g^{[\lambda]}$ is a function that is often non-linear and that maps a value $v_{i^{[\lambda]}}^{[\lambda]}$ to the output activation of a neuron $a_{i^{[\lambda]}}^{[\lambda]}$ (see eq. (2.7))[3]. The basic role of the activation function is to allow a network to learn non-linear relations. Generally, all functions that are static and differentiable could be used as activation function. Monotonic functions have the advantage that no information is lost because of ambiguities of the output values. In the following, the activation functions that are used in this work are defined. The index of the layer λ and the index of the neuron $i^{[\lambda]}$ are omitted in eqs. (2.21) to (2.23) as they do not change. In early approaches, mostly sigmoid functions were used as nonlinearity. Especially the logistic function

$$g(v) = \frac{1}{1 + \mathrm{e}^{-v}}. \tag{2.21}$$

and the hyperbolic tangent

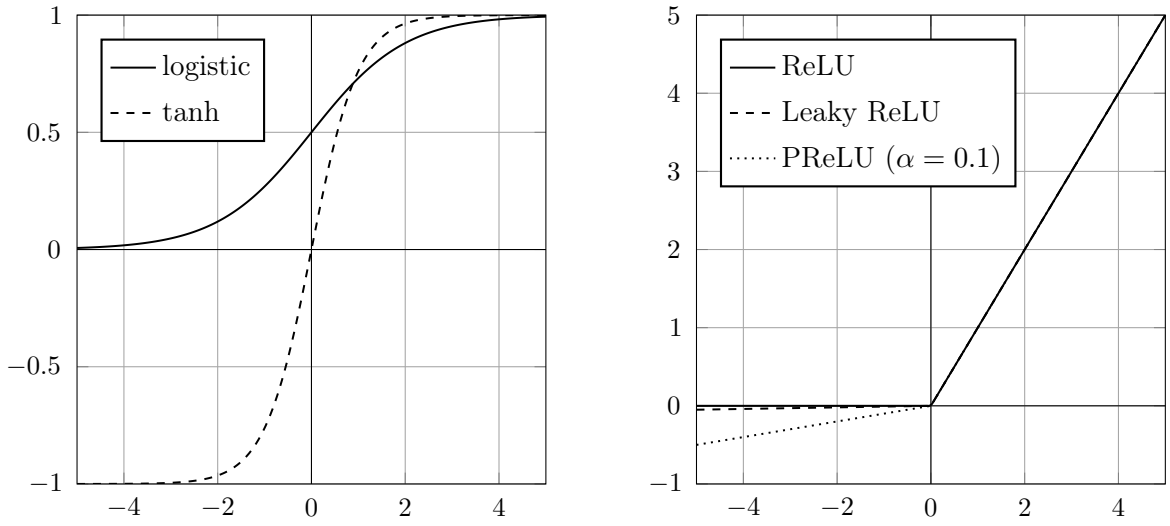
$$g(v) = \frac{\mathrm{e}^{v} - \mathrm{e}^{-v}}{\mathrm{e}^{v} + \mathrm{e}^{-v}}. \tag{2.22}$$

proved to yield good training results. More recently, rectified linear units (ReLUs) and modifications like leaky ReLUs improved the results further on many ML tasks [GBB11; MHN13]. The leaky ReLU function can be formulated as

$$g(v) = \begin{cases} v & \text{for } v > 0 \\ \alpha \cdot v & \text{otherwise} \end{cases}, \tag{2.23}$$

with $\alpha = 0.01$. The basic ReLU function is obtained by setting $\alpha = 0$. A further variant amongst the rectifiers is the parametric rectified linear unit (PReLU), where the factor α is trained together with the weights and biases of the DNN [He+15]. In fig. 2.4, all the previously defined activation functions are depicted.

[3] An activation function can also be described as a function that maps a vector $\mathbf{v}^{[\lambda]}$ to the output activation vector $\mathbf{a}^{[\lambda]}$.



(a) Sigmoid functions

(b) Rectifiers

Figure 2.4: Frequently used activation functions for DNNs. Sigmoid functions (a) have been used for a long time in ML. Today, rectifiers (b) are more commonly used for training neural networks. The variable α in the PReLU activation function is trained like the other network variables. The value of $\alpha = 0.1$ was chosen freely for this figure. Note that there are more well known activation functions, these are just the ones that are important for this work.

A special role has the activation function of the last layer. It defines the range of possible output values of the network with its range of values. In regression networks, the output activation is often omitted, so that the sum of bias and weighted input activations of the last layer $v_i^{[\mathcal{L}]}$ directly yields the predicted output $\hat{y}_i$. For classification tasks, the common activation function at the output is the SoftMax function [GBC16]. It is a normalized exponential function that converts the real numbered vector $\mathbf{v}^{[\mathcal{L}]}$ into a probability distribution

$$g(\mathbf{v}^{[\mathcal{L}]}, i^{[\mathcal{L}]}) = \frac{\mathrm{e}^{v_{i^{[\mathcal{L}]}}^{[\mathcal{L}]}}}{\sum_{\iota=0}^{N_\mathbf{y}-1} \mathrm{e}^{v_\iota^{[\mathcal{L}]}}}. \tag{2.24}$$

2.2.4 Weight Initialization

A common initialization approach is to assign random values to weight factors and zeros to additive bias values. In 2010, Glorot et al. proposed the normalized initialization which is often called Xavier initialization [GB10]. In the Xavier initialization, the weights in every layer $\lambda > 0$ follow a uniform distribution:

$$\mathbf{W}^{[\lambda]} \sim \mathcal{U}\left(-\frac{\sqrt{6}}{\sqrt{N^{[\lambda-1]} + N^{[\lambda]}}}, \frac{\sqrt{6}}{\sqrt{N^{[\lambda-1]} + N^{[\lambda]}}}\right). \tag{2.25}$$

When dealing with DNNs with many layers and the ReLU activation function, this initialization can lead to vanishing or exploding gradients [He+15]. This means that the gradients are either extremely large or very close to zero in the last layers. A solution was proposed by He et al. in 2015 [He+15] by sampling from a normal distribution:

$$\mathbf{W}^{[\lambda]} \sim \mathcal{N}\left(0, \frac{2}{N^{[\lambda-1]}}\right). \tag{2.26}$$

In this work, the Xavier initialization is used without major drawbacks as the number of layers is low for all networks.

2.2.5 Regularization

If the number of neurons in a DNN is too high compared to the amount of training data, overfitting occurs. This means that the network learns to exactly reproduce the training targets without generalizing or learning the principle behind. As a result, a network that was trained with overfitting is expected to give poor prediction results on input data that was not presented to the network in the training process. There are several ways to reduce the amount of overfitting. Besides the options of increasing the amount of training data or decreasing the network complexity, the main means to tackle this problem is regularization. Regularization methods are generally applied in the training stage. Two different regularization methods are used in this work and will be explained in the following: L2-regularization and dropout training.

L2-Regularization L2-Regularization and other similar methods are implemented by adding a regularization term to the loss function. This term usually penalizes high values in the weight matrices in order to reduce strong dependencies on single values. This helps the network to generalize better and to be less sensitive to unseen deviations in single input feature values. L2-regularization might be the most common example and was applied to the trainings in this work. The squared L2-norm of all weights in a network

$$\bar{J}^{\mathrm{reg}}(m, \Theta) = \sum_{\lambda=1}^{\mathcal{L}} \sum_{i^{[\lambda]}=0}^{N^{[\lambda]}-1} \sum_{i^{[\lambda-1]}=0}^{N^{[\lambda-1]}-1} \left| w_{i^{[\lambda]}, i^{[\lambda-1]}}^{[\lambda]} \right|^2 \tag{2.27}$$

$$= \sum_{\lambda=1}^{\mathcal{L}} \left\| \mathbf{W}^{[\lambda]} \right\|_2^2 \tag{2.28}$$

is added to the loss function after being multiplied with a small scaling factor γ^{reg}. Applied to the MSE loss function, which can also be weighted by a factor γ^{mse}, the combined loss function computes to

$$\bar{J}^{\mathrm{mse,reg}}(m, \Theta) = \gamma^{\mathrm{mse}} \bar{J}^{\mathrm{mse}}(m, \Theta) + \gamma^{\mathrm{reg}} \bar{J}^{\mathrm{reg}}(m, \Theta). \tag{2.29}$$

Possible variations include taking the bias values into account and replacing the L2-norm by the L1-norm.

Dropout Training Another class of regularization methods prevents from overfitting by inserting noise into the network. Dropout training [Sri+14] might be the most popular example. In the training process, randomly selected weights, activations or biases are temporarily set to zero. This selection changes after each training sample. Like this, the network does not see exactly the same inputs and internal representations when training for multiple epochs. Like in L2-regularization, the network cannot exclusively rely on single nodes or activations. In the prediction stage, a factor is multiplied to the activations instead of setting a part of the weights to zero.

2.2.6 Time Dependencies in Neural Networks

ABE is, similar to most speech processing tasks, a time-series problem. For each time step, an output vector has to be predicted by a neural network. In most time-series problems, there is a dependency between the current target value and the context in the future and the past. In a real-time application like ABE, the current result has to be predicted based on the current and optionally the previous input vectors. Other applications allow to additionally use future input values in order to predict with a higher accuracy. Three ways shall be noted here that allow to model the dependencies between different time steps: A basic approach is to build a statistical model like an HMM in addition to the DNN that incorporates the time dependencies. However, the neural network can also directly learn these dependencies. Recurrent neural networks (RNNs) were developed exactly for this usecase. The difference to non recurrent networks is that in RNNs, recursive connections can feed information from a node in the network to the same node at the next timestep. With other words, these networks have a memory to store the last values in each node and are able to learn when to store and when to delete a value based on the inputs. The third possibility is to feed the information of the time context to the network in the input features. A simple but expensive method is to stack a small number of input vectors for the last timesteps in order to create a long input feature vector. A variant of this is to calculate the derivative of the input vector over time and to stack this delta feature vector to the current one. The second derivative is called the delta-delta feature vector and can also be stacked to the input features. As the time context with a strong correlation for ABE is rather small, the cheap delta and delta-delta features are used in this thesis. HMMs and RNNs are therefore not explained in more detail.

2.2.7 Multi-Condition Training

The optimal outcome of a DNN training is that the underlying principle is learned by the DNN. However, the trained network often performs better on data that is similar to the training data. Regularization methods can help to generalize better but only to a certain extent. It is therefore beneficial to represent as many relevant input data conditions as possible in the training data. This could be achieved by collecting a huge amount of data that contains a sufficient amount of variations. As this is often

expensive, the variations can also be generated artificially by modifying the present data. This is the principle of multi-condition training. However, when the training data is varied artificially by also representing non-realistic data, the performance of the DNN can decrease caused by a more complex task. Finally, a trade-off between robustness and performance has to be found.

For the purpose of ABE, multi-condition training can be applied in the following dimensions:

- **Microphone and Room**
  The microphone and the room characteristics have a major influence on the mean energy distribution in the frequency domain (FD) which can be represented as a filter or an impulse response. Especially in ABE, where the upper half of the distribution is predicted by the DNN, a robustness regarding different impulse responses is important. The variation can either be achieved by recording speech directly in different rooms and with different microphones or by measuring just the impulse responses of the rooms and microphones. The impulse responses can then be convolved with a clean speech signal that was recorded in an acoustically dry environment in order to simulate the recordings under varying conditions.

- **Loudness Level**
  The predicted energy distribution in the FD should be independent of the signal amplitude. This dependency can be learned by the network when different loudness levels are presented to the DNN. The amplification of the signal with a factor that is randomly chosen between two limits improves the generalization and is easy to implement.

- **Gender, Language, and Speaker's Voice**
  Evaluations showed that the subjective performance of ABE approaches depends on the language of the speech samples [Abe+16]. Accordingly, the characteristics of the speaker's gender and his or her voice could affect the training if it was represented too often in the training data.

- **Equalizer and Recording Environment**
  Depending on the hardware and software in a realization of ABE, the frequency response of the audio system might differ. To take this into account, random equalizers with smooth maxima and minima can be applied to the speech signal.

2.2.8 Multi-Task Learning

In all DNNs presented so far, there was always one single training target, which could be a scalar or a vector. Multi-task learning (MTL) aims at achieving a better generalization by training multiple correlated tasks at the same time with the same DNN [Car97]. The MTL loss function is defined as a weighted sum of one loss function per task. Regarding ABE, MTL can be applied by training a classification that detects whether the current

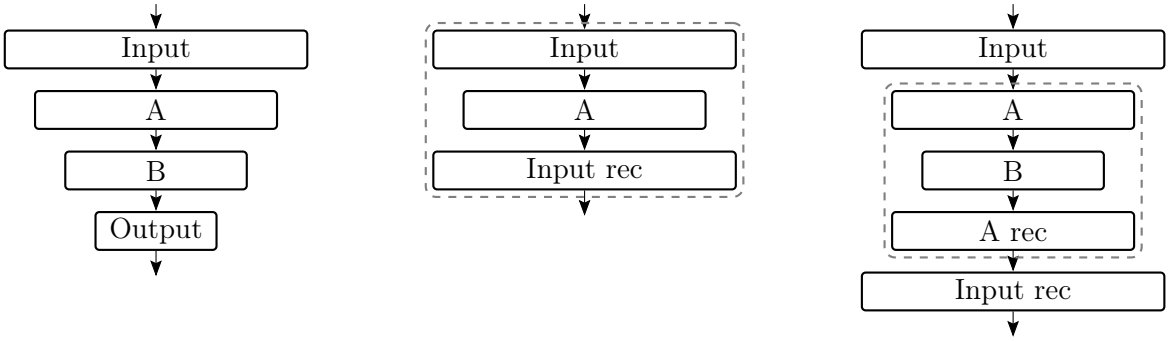
frame contains speech in parallel to the regression on the energy prediction task. The tasks have to be chosen carefully, as some tasks might have conflicting targets so that both tasks cannot be completely solved with the same network [Car97]. Different types of MTL can be categorized depending on the partitioning of the network in layers that are used for both tasks, also called shared layers, and layers that are specific for one of the tasks. In this thesis, all layers except for the output layers were shared layers when MTL was applied.

2.3 Unsupervised Pre-Training

After or instead of initializing a DNN with random values, the network can be initialized by unsupervised pre-training. The goal of this method is to find good inner representations of the input data independent of the target. This can partly prevent the network from converging to local minima and from losing parts of the information in every layer. If a part of the input information is lost once propagating through the network in a standard feedforward architecture, it cannot be used for the prediction any more.

Unsupervised pre-training can be achieved by training auto-encoder networks. An auto-encoder is a DNN that has the same input and target values [Erh+10]. It consists of an encoder part, in which an inner representation of the input data is created, and a decoder part, that aims at reconstructing the inputs from this inner representation. Consequently, the network has an input layer, a hidden layer with the inner representation and an output layer. While a pure reproduction might seem to be very easy, there are some methods that make the auto-encoder robust against noise or data errors. These methods include adding noise on the input, making the hidden layer small, and applying regularization methods.

Stacked auto-encoders are used when multiple layers of a DNN shall be initialized [Vin+10]. For each hidden layer in a DNN that shall be pre-trained, a single auto-encoder is created. The first auto-encoder is generated by connecting the output of the first hidden layer to a copy of the input layer. After having trained this auto-encoder, the hidden layer is replaced by the second auto-encoder, which consists of two versions of this first hidden layer with the second hidden layer in between. An example is depicted in fig. 2.5. Depending on the implementation, the weights of the first auto-encoder can be kept fixed when the weights of the second one are trained. Like this, the whole DNN can be pre-trained in an iterative process. Finally, the whole part that reconstructs the input is not needed any more and the encoder part is connected to the real output. Stacked auto-encoders were applied as a pre-training step for some of the DNN trainings that were trained in the scope of to this work.



(a) DNN (b) Auto-encoder 1 (c) Auto-encoder 2

Figure 2.5: Pre-training of a DNN with a stacked auto-encoder. The layers are represented by the boxes, of which the first at the top is the input layer without weights. All other layers contain the weight matrix that is multiplied on the outputs of the preceding layer. The dashed boxes around the layers represent the auto-encoders. The DNN that shall be pre-trained, is shown in (a). The first auto-encoder, which trains the weights in layer A, is depicted in (b). The second auto-encoder, which is built by replacing layer A in auto-encoder 1 by three layers, is given in (c). After training both auto-encoders, the weights for layers A and B can be used as initialization for the DNN. The layers with the suffix 'rec' are meant to reconstruct the information of the respective layers.

2.4 Input Feature Selection Methods

DNNs are powerful in learning a mapping from input features to target features. For high-dimensional input or target data, feeding the raw data into the network is not cost-efficient because of the large layers at the input or output of the network. Finding a compact representation of the data and feeding this to the network is a possible solution for this problem. While this might work well for the target feature, the network might need details that are not contained in the compact representation for good prediction results. These details can be provided to the DNN by additional input features. This section deals with the selection of these input features.

Feature selection methods can be categorized into supervised and unsupervised methods. Supervised methods require a supervised training where the targets are known for the whole training dataset. As this is the case in the DNN trainings that were performed in the scope of this thesis, only supervised methods are handled in the following.

Supervised feature selection can be implemented by using *filter*, *wrapper*, or *embedded* models [TAL14]. *Filter models* (e.g. mutual information) evaluate the similarity of the input data. They do not take into account with which kind of algorithm the targets shall be extracted from the input features [TAL14]. A feature selection approach for ABE based on mutual information and a separability measure was presented in [JV04]. However, it was applied to a classification task where the output feature is a probability

vector for WB spectral envelopes that are stored in a codebook. In the current approach, a regression DNN is trained to estimate the spectral envelope directly. Therefore, the separability measure cannot be used in this case. The applicability of mutual information for the feature selection for neural networks was tested in [Bat94]. It was stated that the mutual information could give a lower bound of the convergence error so that unnecessary features could be found immediately. It might still not be the best approach to find a good feature set, as the performance of a DNN with the selected feature set is not taken into account [KJ97]. *Wrapper models* utilize a model (e.g. a DNN) that shall map a given input feature set to target features in order to evaluate the quality of the feature set [TAL14]. The main drawback of these models is the low efficiency for large feature pools where many models have to be trained. *Embedded models* do the training of the network and the feature selection simultaneously. They form a trade-off between a good selection performance and a high efficiency [TAL14]. The only drawback is that the implementation is more complex.

2.5 Generative Adversarial Networks

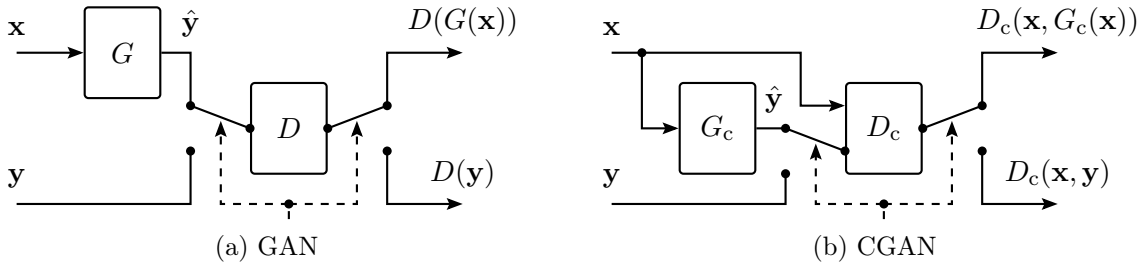
Generative adversarial networks (GANs) estimate generative models by capturing a data distribution in a DNN [Goo+14]. Random noise $\mathbf{z}$ is given as input to a DNN that shall generate a data sample that fits to a given data distribution. This is achieved by training two adversarial networks: the generator network G and the discriminator network D (see fig. 2.6a). While G learns to generate data $\hat{\mathbf{y}}(l, \Theta^G)$ that seems to be real data $\mathbf{y}(l)$, D learns to distinguish this generated data from real data. Like this, the discriminator's knowledge of the differences between generated and real data helps the generator to improve the prediction. The training process can be interpreted as a game where both players have to improve more and more in order to win. The optimal training result after convergence is that the generator produces samples that cannot be distinguished from real data, in other words, the generator finally wins. In this thesis, the random noise at the input of G is replaced by the NB input $\mathbf{x}(l)$ in order to generate specific output data.

D classifies whether its input is true or estimated by G. Correct decisions would be $D(\mathbf{y}(l)) = 1$ and $D(\hat{\mathbf{y}}(l, \Theta^G)) = 0$. The optimization rule for D can be formulated as the maximization of the following loss function [Iso+16]:

$$J^{\text{gan}}(l, \Theta^G, \Theta^D) = \log(D(\mathbf{y}(l), \Theta^D)) + \log(1 - D(\hat{\mathbf{y}}(l, \Theta^G), \Theta^D)) \tag{2.30}$$

$$= \log(D(\mathbf{y}(l), \Theta^D)) + \log(1 - D(G(\mathbf{x}(l), \Theta^G), \Theta^D)). \tag{2.31}$$

When taking into account that the output of D lies between 0 and 1 and that the logarithm is a monotonically increasing function, which is negative for values smaller than 1, it can be concluded that the loss is zero in case all predictions are correct and negative otherwise. In opposite to that, the optimization rule for G is to minimize eq. (2.31). This means that the objective for G is to estimate data that D would classify



(a) GAN (b) CGAN

Figure 2.6: Differences between GAN (a) and CGAN (b) training. G and G_c denote the generator network and D and D_c denote the discriminator network. The training of the discriminator is done alternately either on real target data $\mathbf{y}(l)$ or on data predicted by the generator $\hat{\mathbf{y}}(l)$. The generator is just trained on input data $\mathbf{x}(l)$ when the switch is in the depicted position. Note that the frame index l and the network parameters Θ are omitted in the figures and that the discriminators in the upper and lower graphs contain the same variables.

as real data. The networks G and D are never trained simultaneously but sequentially. It is an important task to find a suitable number of training updates of D and G that shall be calculated in one period until the other network is trained again. This has a high effect on the stability of the training. With a randomly chosen number of updates for G and D, one of the networks might dominate the game from the beginning and the training might diverge.

CGAN The standard GAN generates data just from the random noise input $\mathbf{z}(l)$, which is here replaced by the input $\mathbf{x}(l)$. The discriminator D will then judge the prediction of G without knowing the input $\mathbf{x}(l)$. The consequence for the training of G is that the loss function does not directly depend on the input data. The conditional GAN (CGAN) [MO14; Iso+16] architecture takes this into account by additionally feeding the input $\mathbf{x}(l)$ to the discriminator D. Like this, D can make the decision whether the data seems to be real or generated based on the given input $\mathbf{x}(l)$ of the generator. In order to separate the CGAN from the basic GAN approach in the equations, the networks G and D are called G_c and D_c when they belong to a CGAN. The difference between GAN and CGAN is shown in fig. 2.6b. The only difference in the training is the modification of the loss function from eq. (2.31) with the additional dependencies [Iso+16]:

$$J^{\mathrm{cgan}}(l, \Theta^G, \Theta^D) = \log(D_\mathrm{c}(\mathbf{x}(l), \mathbf{y}(l), \Theta^D)) + \log(1 - D_\mathrm{c}(\mathbf{x}(l), G_\mathrm{c}(\mathbf{x}(l), \Theta^G), \Theta^D)). \tag{2.32}$$

2.6 Hyperparameters

Besides the parameters Θ of a network, there are many parameters that control the training process, called hyperparameters [GBC16]. They have to be set accurately in order to achieve good training results and they are not modified by the training algorithm. There are also approaches for an automatic hyperparameter search [BB12], but they come along with high computational costs. In this section, some parameters that are relevant for this work are described briefly.

Batch Size The (mini-) batch size is one of the parameters that has a strong influence on the convergence and the performance. Choosing a low batch size may help with a good convergence as it tends to pure SGD, where every sample is directly evaluated. However, it takes a long time to train because the gradient has to be calculated for each sample. Setting the batch size to high values increases the required processor memory and is therefore only possible in a limited range [Smi18].

Number of Layers Deep networks with many hidden layers are theoretically able to learn highly complex causal connections. However, they are also harder to train than DNNs with less layers. The experience gained from the trainings for ABE showed that more layers do not always increase the final performance. Finally, the best results were achieved using one to three hidden layers.

Size of Layers There are no constraints in choosing a network architecture for a given problem. Besides the high flexibility, this makes it difficult to find a reasonably good architecture. Although many specific types of DNN shapes were investigated, most of the results are only valid for the respective task. It seems that there is still no good method to find the optimal architecture except for training models with multiple architectures and comparing their performance.

Learning Rate/Step Size The learning rate η sets the size of the updates that are performed on the weights in a network. Most current optimizers adapt the learning rate dependent on the convergence state. By setting the initial learning rate, the speed and the final state of the convergence can be influenced. Again, a value has to be found experimentally. A very small value can cause overfitting while large values can lead the training process to diverge [Smi18]. In the trainings related to this work, learning rates between 10^{-6} and 10^{-4} yielded the best results.

Chapter 3

Speech Production and Transmission

For a better understanding of the challenges that come along with the development of a high-performing ABE solution, basic knowledge of speech production and speech transmission is necessary. Regarding the topic of speech production, the types of sounds in human speech and their characteristics are investigated in this chapter. Differences in the speech production let us distinguish between several classes of speech sounds. These classes also differ in their mean energy distributions over the frequency spectrum. The amount of energy that is located in the UB for a specific phoneme class can, on the one hand, be used in the evaluation of ABE algorithms. On the other hand, the training can be modified to focus on the correct prediction for speech sounds with high UB energy.

In the part about speech transmission, some characteristics of a speech signal, which was transmitted through a telephone network, are investigated. The amount of data that is transmitted shall generally be minimized in order to obtain a high performance at low costs. This was one of the reasons for transmitting speech at 8 kHz when the NB analog telephone standard was developed [Jax02]. With the ability to transmit speech at 16 kHz, the quality of telephone speech could be improved significantly. But not only the bandwidth affects the quality of a telephone speech signal. Encoding the signal at low bitrates allows for a minimal traffic but at the same time reduces the quality of the decoded speech signal at the receiver's side. An ABE algorithm should yield good results for all different types of codecs that are used in the transmission of NB speech. When dealing with DNN-based ABE, this means that these codecs should be present in the training data so that the network can adapt to the respective characteristics.

This chapter is structured in two parts: In a first part, the topic of speech production is outlined in section 3.1 and a way how this process can be modeled as a system in the field of signal processing is presented in section 3.2. The second part deals with the transmission of speech signals: Section 3.3 gives a brief overview of the common bandwidths and codecs that are used in telecommunication systems, including the definitions of the terms NB and WB. Alternative representations of a speech signal like the short-term Fourier transform (STFT), the mel-scale and the cepstrum are explained in section 3.4.

3.1 Basics of Speech Production

This section gives a brief overview of how speech is produced and how speech sounds can be categorized. Human speech can be subdivided into its different speech sounds, which are called phonemes. The phonemes are categorized into phoneme classes depending on the way of articulation. All human speech sounds are produced by an air stream, mostly generated by a contraction of the lungs, as it is shown in the model in fig. 3.1b. In this case, the air flows through the vocal tract and exits the mouth or the nose. As a reference, the anatomic scheme in fig. 3.1a shows a more detailed version of the vocal tract.

The sound source of a speech sound is called its excitation. The generated sound is mostly modified by the resonances of the different cavities in the vocal tract until it is radiated at the mouth or the nose. This can be modeled as a filter in system theory. The source-filter model of speech production that is based on this subdivision is described in section 3.2.

For a first classification of phonemes, the excitation of speech sounds is investigated. Three different types of excitation of human speech sounds can be distinguished: *voiced* excitation, *unvoiced* excitation, and *transient* excitation. All of these excitation types are described in the following list:

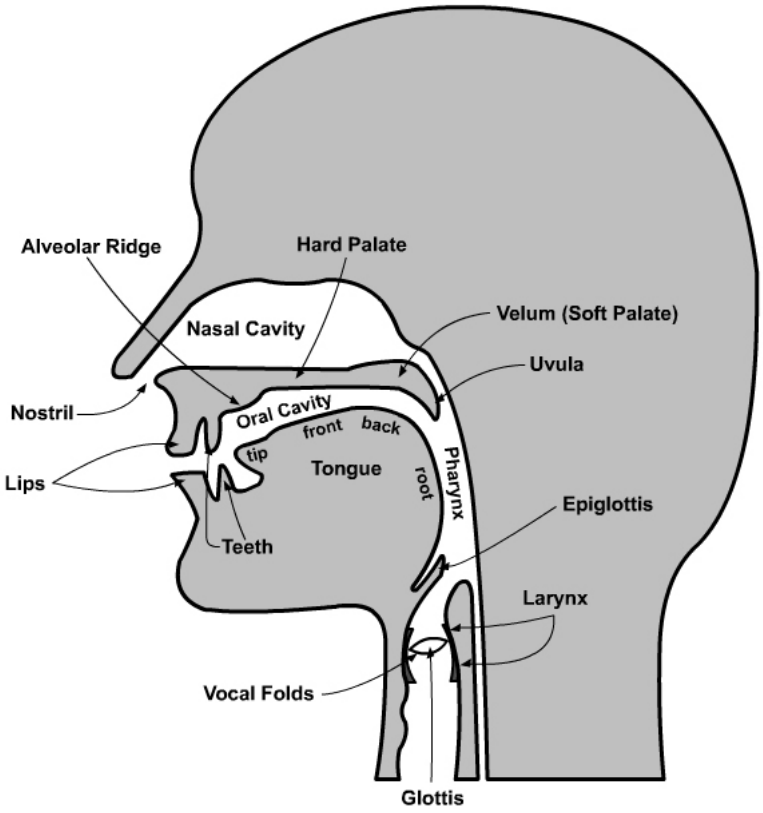
- **Voiced Excitation**
  When the vocal folds are tensed, the air stream lets the vocal folds oscillate quasi-periodically. In every period, the closed vocal folds are opened by the rising subglottal pressure originating from the air stream and closed again, partly because of the falling pressure and the tension in the vocal folds[1]. These sounds are called voiced sounds. The frequency of the quasi-periodic oscillation is called the fundamental frequency or pitch frequency f_0. It is the frequency at which speech sounds are perceived. The spectrum of voiced sounds contains a peak at the fundamental frequency and additional peaks at integer multiples of f_0.

- **Unvoiced Excitation**
  If the vocal folds are opened, the air stream can pass without causing an oscillation. Consequently, this class of sounds is called unvoiced. The sound is mostly created somewhere in the vocal tract by turbulences, depending on the phoneme.

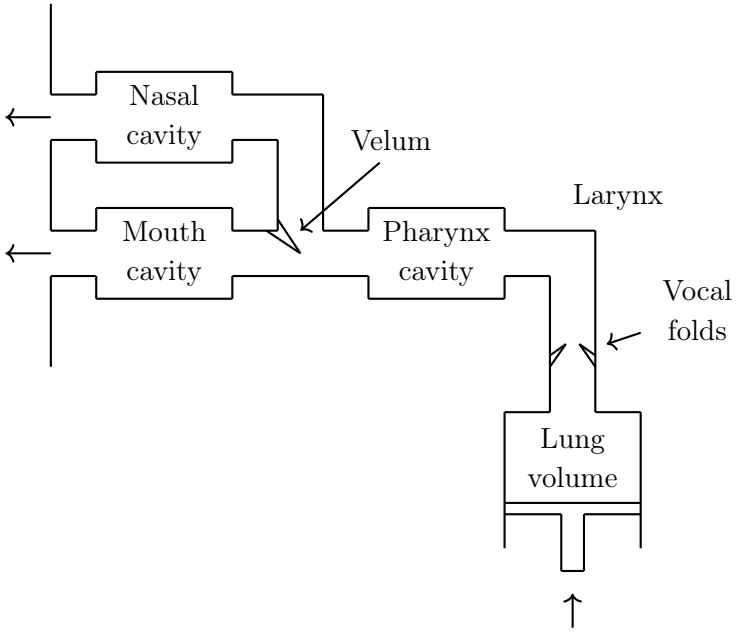
- **Transient Excitation**
  If there is a complete blocking of the air stream for a short time which is then released, the occurring sound is called transient. The class of transients contains plosives like, e.g., [p], [t], [k], [b], [d], and [g].

Voiced and unvoiced sounds can further be specified based on the shape of the vocal tract. Obstructions in the vocal tract that hinder the air from flowing constantly create turbulences which generate noisy sounds. For narrow obstructions, the generated noise

[1] This is just a brief explanation, more details are given in [LM96].



(a) Human vocal tract, adapted from [UCL18]



(b) Abstract model of the vocal tract, adapted from [RSS10]

Figure 3.1: The human vocal tract, depicted as anatomic scheme (a) and as abstract model (b).

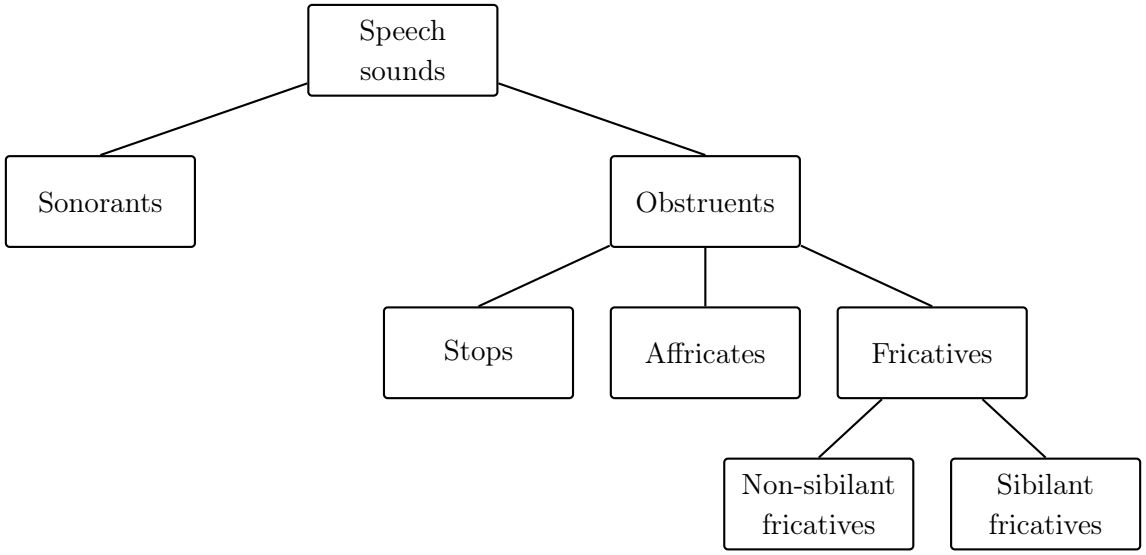


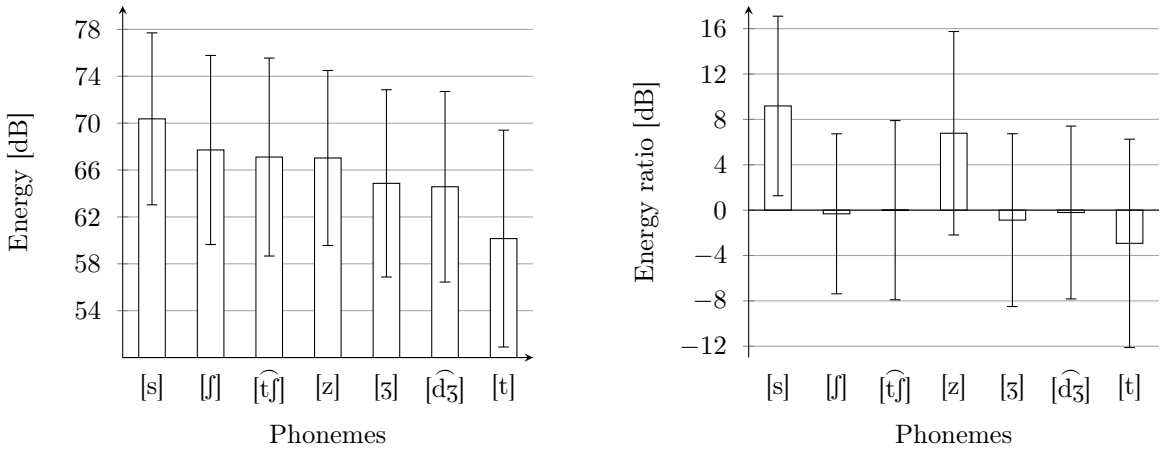
Figure 3.2: Subdivision of different speech sounds.

dominates the sound. This class of sounds is called *obstruents*. All other sounds, where the oscillating vocal folds dominate the sound, are called *sonorants*. The categories of speech sounds that are distinguished in the following are depicted in a diagram in fig. 3.2.

Another general classification of phonemes distinguishes between vowels and consonants. A vowel is originally defined as a syllabic sound without obstructions in the vocal tract, where a syllabic phoneme is a phoneme that can form a syllable on its own [LM96]. All other sounds are consonants, which can be further specified by the *manner of articulation* and the *place of articulation* in the vocal tract. As the definition of vowels is not only related to the way of articulation of the speech sound, vowels will not form a separate class of phonemes in this chapter.

The main energy of sonorants is typically located in low frequencies around the fundamental frequency and the first harmonics (see fig. 3.4). The local maxima in the spectral envelope that are caused by the resonances in the vocal tract are called formants. This thesis focuses on ABE towards higher frequencies, which means that especially the energy above 3.4 kHz, which has to be predicted by the algorithm, is important. As the main energy of sonorants is concentrated in low frequencies, the energy that would be introduced by an ABE algorithm is relatively small. Therefore, all sonorants will be considered as one phoneme class in the following without further categorization.

The class of obstruents is further subdivided into the classes of *stops*, *fricatives*, and *affricates*. Stops are obstruents with a transient excitation, fricatives have a voiced or unvoiced excitation, and affricates are combinations of one stop and one fricative [LM96]. Two classes of fricatives can be further distinguished based on the manner of articulation: sibilant and non-sibilant fricatives. *Sibilants* are produced by directing a fast, turbulent stream of air with the tongue towards an obstacle (like the teeth) [LM96]. The English



(a) $p_{\mathrm{ub}}^{[\mathrm{dB}]}$ (b) $p_{\mathrm{ub}}^{[\mathrm{dB}]} - p_{\mathrm{nb}}^{[\mathrm{dB}]}$

Figure 3.3: Comparison of the mean energy of phonemes in the TIMIT dataset in dB (the speech files were read as 16-bit integer values). Just the seven phonemes with the highest energy in the UB (above 3.4 kHz) are shown. The comparison results are shown for the UB energy (a) and for the ratio between the UB energy and the NB energy between 0.3 and 3.4 kHz (b). The error bars represent the standard deviation towards lower and higher values.

sibilant fricatives are [s], [z] (voiced [s]), [ʃ] (in English sh), and [ʒ] (voiced [ʃ]). They form the only class of sounds that contains a higher energy above 4 kHz than below on the TIMIT dataset in fig. 3.5. *Non-sibilant* fricatives are also produced by a fast air stream, but in this case the sound is caused by the turbulences at the constriction directly [LM96].

As stated above, the main interest lies on speech sounds that have a high amount of energy in the UB. An analysis of the TIMIT dataset [Gar+93], which is transcribed precisely, yielded the mean UB energy of different phonemes. In fig. 3.3a, the mean UB energy of the seven phonemes with the highest UB energy is shown. Figure 3.3b depicts the ratio between the UB and the NB energy. The only phonemes that have a reasonably higher energy in the UB than in the NB are [s] and its voiced version [z]. It is also noticeable that the six phonemes with the highest UB energy coincide with the list of sibilant fricatives and the respective affricates.

The energy distribution of non-sibilant fricatives was found to be slightly similar to the energy distribution of stops in TIMIT. Additionally, the UB energy is rather low. Consequently, these classes do not have to be separated and were combined to the class of non-sibilant obstruents for the following evaluation. In fig. 3.4, the mean spectra of the four phoneme classes, sibilant fricatives, sibilant affricates, non-sibilant obstruents, and sonorants, are depicted. The graphs show once more the high UB energy of all sibilant obstruents. The energy in the mean spectra of different phoneme classes for the NB and the UB is shown in fig. 3.5. This data can be a basis for a classification in different phoneme classes that can later be treated differently in the training of a

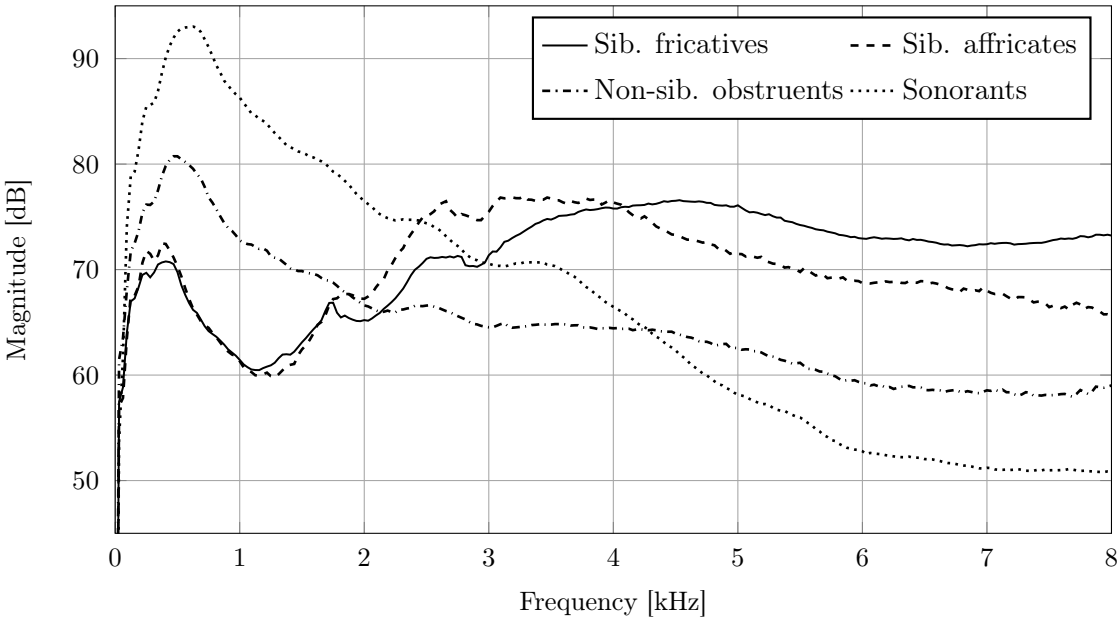


Figure 3.4: Mean spectra of different phoneme classes.

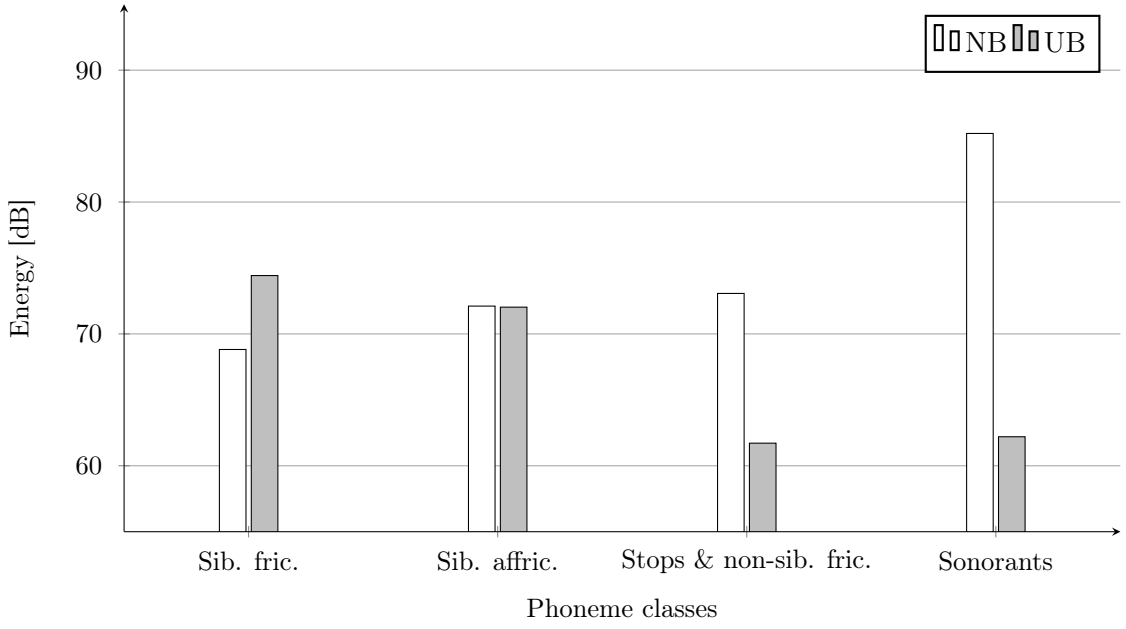


Figure 3.5: Mean energy of different phonemes in the NB and the UB for the TIMIT dataset.

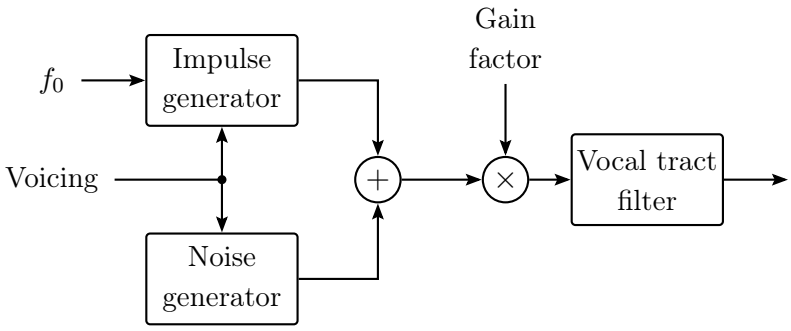


Figure 3.6: Source-filter model of speech production. Generally, for voiced speech, the impulse generator is used and for unvoiced speech, noise is generated. The combination of both is also possible as both generators can produce signals at arbitrary levels independently of each other. The vocal tract is modeled as digital filter.

DNN in order to achieve good results. It has been a common problem that many ABE systems did not introduce enough energy in the UB for [s] and [z] [BAF14]. The graphs make clear why this task is so complicated: Based on the small amount of energy in the NB, the algorithm has to predict the high amount of energy in the UB.

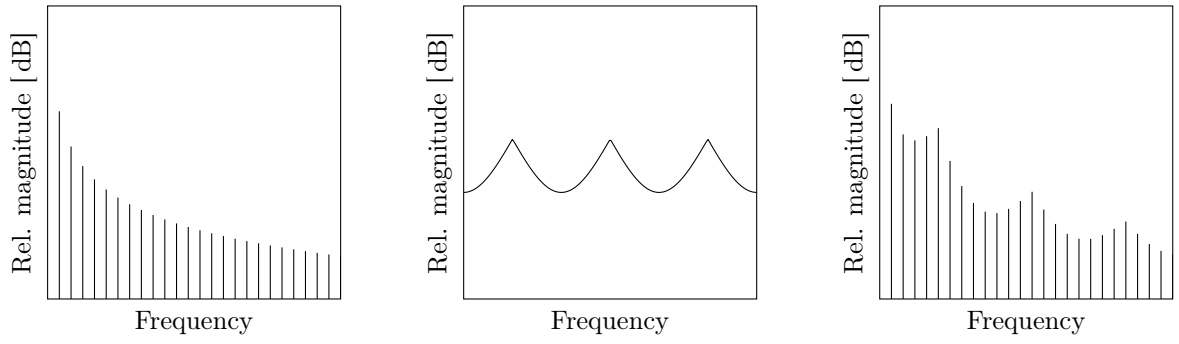
3.2 Source-Filter Model of Speech Production

Speech can be described as a superposition of multiple sound sources with a time-variant modification caused by the vocal tract, the filter. This section describes the source-filter model of speech production [Fan60] in more detail and shows up how this can be used for ABE.

An example for a voiced sound source are the oscillating vocal folds and possible unvoiced sound sources are narrow obstructions that cause turbulences. The superposition of voiced and noisy sounds is also possible like it is the case for voiced fricatives. The class of transient excitation is not considered explicitly in the original model, which distinguishes between voiced and noisy sound sources [Fan60]. The locations of the unvoiced sound sources differ in the range from the vocal folds to the teeth (dental) and the lips (labial). More locations in the vocal tract are named in fig. 3.1a.

The vocal tract as a whole can be modeled as time-variant filter that modifies the sounds produced by the sound sources. The filter is described by the diameters of the vocal tract from the vocal folds to the lips. It can be modeled as a series of connected tubes with a small length in order to be able to calculate the acoustic transfer function. An example for an ABE system that is based on the tube model is given in [SJV18].

All these phenomena can be modeled in a rather simple way following the source-filter model of speech production [Fan60], which is depicted in fig. 3.6: The unvoiced sound sources are generated by white noise and the voiced sound source is generated by an impulse train. The frequency of these impulses is the fundamental frequency of the speech f_0. A gain factor represents the differences in the loudness of the voice. The



(a) Voiced source (b) Filter (c) Speech signal

Figure 3.7: Schematic example of the source-filter model of speech production for a voiced speech sound. The voiced sound source (a) is filtered with the characteristics of the vocal tract (b) and produces the speech signal (c).

influence of the vocal tract is modeled in a linear, time variant filter that is applied to the generated sounds. In the frequency spectrum, the filter can be understood as a spectral envelope that controls the energy distribution. The combined source signals build the excitation signal. An example for a voiced sound source is depicted in fig. 3.7.

3.3 Bandwidth of Speech Signals

The bandwidth of analog telecommunication systems is defined by the International Telecommunication Union (ITU)-T recommendation G.120 [ITU98] to the bandpass (BP) between 300 Hz and 3.4 kHz. This bandwidth is called NB. The specification was intended to have low data traffic while maintaining about 99% intelligibility for sentences of clean telephone speech [Jax02]. The basic speech transmission in digital cellular networks is defined in ITU-T rec. G.711. The sampling rate is set to 8 kHz in these pulse code modulation (PCM)-based transmission systems [ITU88a]. This results in a cutoff frequency lower than 4 kHz. The attenuation constraints for PCM-based transmission systems are normed in ITU-T rec. G.712 [ITU01]. These constraints are similar to those defined for analog telephony and are also based on the BP between 300 Hz and 3.4 kHz. A general requirement for the attenuation characteristics for national networks is defined in ITU-T rec. G.120 [ITU98] and is shown in fig. 3.8.

Nowadays, most telephone systems support WB calls with a bandwidth that ranges from 50 Hz up to 7 kHz [ITU12]. WB quality is also called *HD Voice* in consumer products. A requirement for the attenuation characteristics of the WB sending path of hands-free cellphones is defined in ITU-T rec. P.341. It is depicted in fig. 3.9. A standardized characteristics [ITU11a] that fits in these requirements will be used later for simulation purposes. The part of the WB spectrum above the NB bandwidth that is estimated in ABE is called UB in this thesis. The part below 300 Hz that is missing

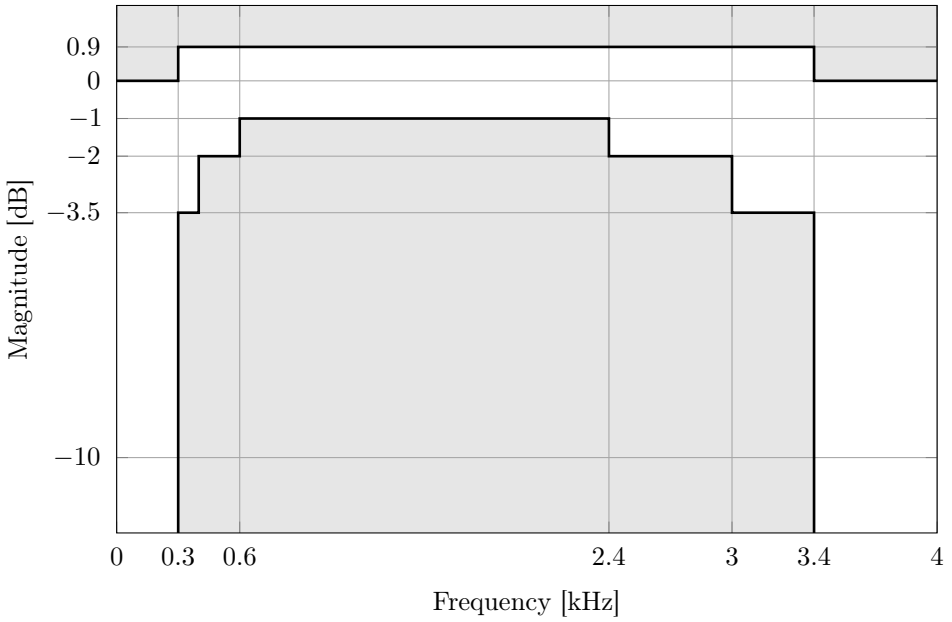


Figure 3.8: Attenuation limits for circuits with 4-kHz channel equipment, adapted from ITU-T rec. G.120 [ITU98].

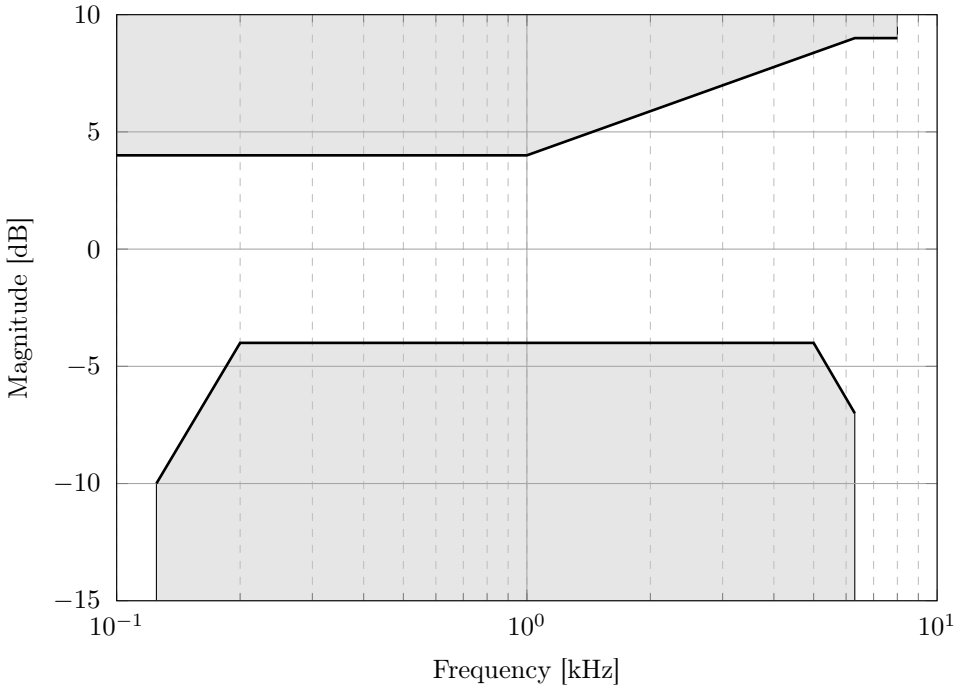


Figure 3.9: Attenuation limits for the hands-free sending path of cellphones, adapted from ITU-T rec. P.341 [ITU11a].

in NB speech can also be recreated by ABE methods. As there are already satisfying algorithms with low complexity for this task [IMS08], this thesis only deals with the extension to higher frequencies.

Even higher bandwidth definitions are super-wideband (SWB) (up to 16 kHz) and fullband (FB) (up to 20 kHz, the full frequency range of human hearing). The high bandwidths SWB and FB are not covered in this thesis.

3.4 Speech Signal Representations

The way a speech signal is represented can be chosen depending on the requirements that are given by each application. The basic representation as waveform has advantages for low-delay applications because every sample can directly be processed. Although ABE is an application where the delay has to be low, processing speech signals in the frequency domain (FD) offers a wide range of possibilities regarding more advanced methods and is therefore applied in this thesis. The possibility to combine ABE with a FD-based noise reduction is another advantage of the spectral processing. The transformation to the FD is implemented as STFT, which is described in section 3.4.1.

After having separated the short-term spectrum into a spectral envelope and an excitation signal, both parts are processed in the ABE algorithm. In the FD, the spectral envelope of a short-term spectrum can be easily obtained by a smoothing of the magnitude in frequency direction. For the spectral envelope, different representations can be chosen. In section 3.4.2 and section 3.4.3, two frequently used representations are explained. These are also applied in the ABE algorithm.

3.4.1 Short-Term Spectrum

In order to process the spectrum of an acoustic signal in real-time, an STFT is applied to the time-domain (TD) signal. The STFT transforms the discrete TD signal to complex-valued short-term spectra in the FD.

The TD signal $s(n)$ is buffered into overlapping frames of N samples with a frame-shift of F samples. Each block is multiplied with an analysis window $w_{\text{ana}}(n)$. The windowed speech signal is transformed by a discrete Fourier transform (DFT) to the complex short-term spectrum

$$S(k,l) = \sum_{n=lF}^{lF+N-1} w_{\text{ana}}(n-lF) \cdot s(n) \cdot \mathrm{e}^{-\mathrm{j}\frac{2\pi k}{N}(n-lF)}, \tag{3.1}$$

with $k \in \{0,\ldots,K-1\}$ and $K = {}^{N}/_{2}+1$. The signal can be described by its magnitude and phase like any complex value. Many tasks in speech processing just modify the magnitude of the short-term spectrum and not the phase. Therefore, some frequently used variants of the representation as magnitude spectrum are given in the following equations. Taking the squared absolute value of the complex spectrum $S(k,l)$ gives the

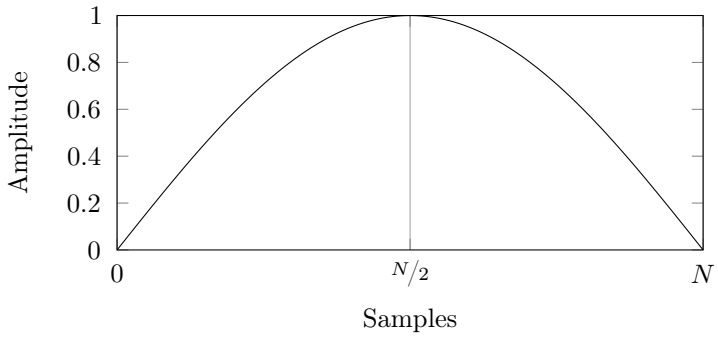


Figure 3.10: Analysis window $w_{\text{ana}}(n)$ that is used in the filterbank. The synthesis window $w_{\text{syn}}(n)$ is identical. It is a square-root Hann window that leads to perfect reconstruction with an overlap of 50%.

real-valued power spectrum (PS)

$$\Phi_{SS}(k,l) = |S(k,l)|^2. \tag{3.2}$$

The amplitude of sound is perceived on a logarithmic scale. The logarithmic amplitude of acoustic signals is usually given in dB. The power spectrum in dB can be calculated according to

$$\Phi_{SS}^{[\text{dB}]}(k,l) = 10\log_{10}(\max(1, \Phi_{SS}(k,l))). \tag{3.3}$$

Taking the maximum with 1 restricts the results of the logarithm to be greater or equal to zero. This dynamics is meant to be sufficient here as the peak level for normalized signals with integer values represented by 16 bit usually lies above 100 dB. Like this, the logarithm can also be calculated when the PS is exactly zero in some time-frequency bins. For those tasks that depend on a single level for a given frame, the power in all frequency bins is added up to the logarithmic power level

$$p^{[\text{dB}]}(l) = 10\log_{10}\left(\max\left(1, \sum_{k=0}^{K-1} \Phi_{SS}(k,l)\right)\right). \tag{3.4}$$

The synthesis of a short-term spectrum to a waveform is achieved by an inverse discrete Fourier transform (IDFT) followed by an overlap-add algorithm, that multiplies every frame with the synthesis window and adds the shifted frames up to one signal. To achieve this, the synthesis window $w_{\text{syn}}(n)$ is defined to be zero for all $n < 0$ and $n \geq N$. The synthesized waveform can then be written as

$$s(n) = \sum_{l=0}^{L_{\text{tot}}-1} w_{\text{syn}}(n - lF) \cdot \frac{1}{N} \cdot \sum_{k=0}^{K-1} S(k,l) \cdot \mathrm{e}^{\mathrm{j}\frac{2\pi k}{N}(n-lF)}, \tag{3.5}$$

where L_{tot} is the number of frames in $S(k,l)$.

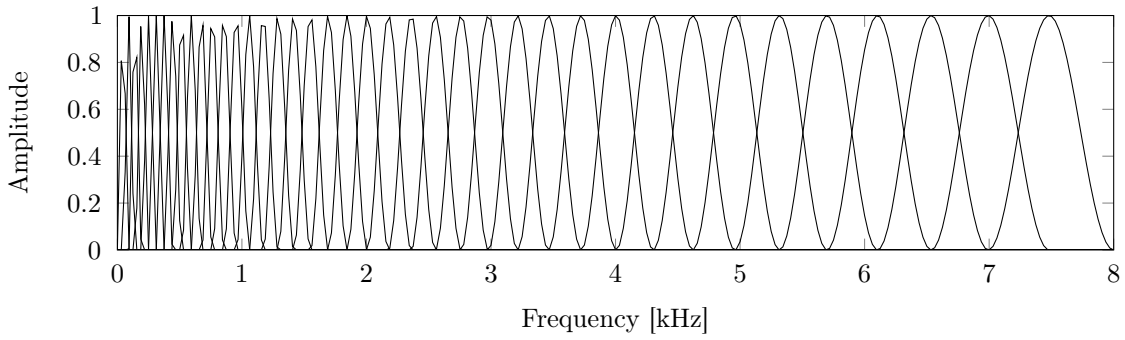


Figure 3.11: Superposition of all filters of the mel filterbank that was used in this thesis. Each filter is formed by a single sine wave instead of a rectangle, which is often done in literature, for smoother transitions. The filters are normalized to a sum of 1 for the conversion to a mel spectrum. For the conversion back to the FD, the filters are not normalized, as displayed in the figure.

3.4.2 Mel Spectrum

The mel filterbank is used to transform a frequency spectrum with uniformly distributed frequency bands to a perceptually motivated frequency scale [SVN37]. The bands of the mel scale get wider towards higher frequencies in a logarithmic manner (see fig. 3.11). This adapts to the human hearing where the perceived frequency also approximately follows a logarithmic scale [SVN37]. A spectral envelope can therefore be efficiently represented by a mel spectrum yielding a strong reduction of the coefficients. The frequency in mel can be calculated from the frequency in Hz with the following conversion formula:

$$f' = 1127 \ln\left(1 + \frac{f}{700}\right), \tag{3.6}$$

where $(\cdot)'$ is used to indicate that a variable is based on mel filter coefficients. The mel-frequency PS of the WB spectrum computes to

$$\Phi_{S'S',\mathrm{wb}}(k', l) = \frac{1}{\sum_{k=0}^{K-1} \mathcal{M}(k', k)} \sum_{k=0}^{K-1} \mathcal{M}(k', k) \cdot \Phi_{SS,\mathrm{wb}}(k, l), \tag{3.7}$$

where k' is the discrete mel band index and $\mathcal{M}(k, k')$ is the transformation coefficient.

The transformation of a mel spectrum back to a spectral envelope in the FD can be achieved with the same parameters but without the normalization term:

$$\Phi_{SS,\mathrm{wb}}(k, l) = \sum_{k'=0}^{K'-1} \mathcal{M}(k', k) \cdot \Phi_{S'S',\mathrm{wb}}(k', l). \tag{3.8}$$

3.4.3 Mel-Frequency Cepstral Coefficients

MFCCs are a compact description of the mel-frequency PS with uncorrelated coefficients. Applying a discrete cosine transform (DCT) of type 2 to the logarithmic mel-based PS

causes the decorrelation and leads to the MFCCs

$$\begin{aligned} c_{\mathrm{nb}}(i_c, l) &= \mathrm{DCT}(\Phi^{[\mathrm{dB}]}_{S'S',\mathrm{nb}}(k', l)) \\ &= \sum_{k'=0}^{K'-1} \Phi^{[\mathrm{dB}]}_{S'S',\mathrm{nb}}(k', l) \cdot \frac{1}{\sqrt{1+\delta(k')}} \cdot \cos\left(\frac{\pi}{K'}\left(k'+\frac{1}{2}\right) i_c\right), \end{aligned} \tag{3.9}$$

where $i_c = \{0, \ldots, N_c - 1\}$ is the MFCC index, $\delta(k')$ is the Kronecker function that is 1 for $k' = 0$ and 0 otherwise. $\Phi^{[\mathrm{dB}]}_{S'S',\mathrm{nb}}(k', l)$ is calculated as the squared NB mel-spectrum, according to eq. (3.2). The conversion back to the mel spectrum is achieved by the inverse discrete cosine transform (IDCT), which is in this case equal to the DCT of type 3:

$$\Phi^{[\mathrm{dB}]}_{S'S',\mathrm{nb}}(k', l) = \sum_{i_c=0}^{N_c-1} c_{\mathrm{nb}}(i_c, l) \cdot \frac{1}{\sqrt{1+\delta(i_c)}} \cdot \cos\left(\frac{\pi}{N_c}\left(k'+\frac{1}{2}\right) i_c\right). \tag{3.10}$$

3.4.4 Codecs for Speech Transmission

Several codecs were developed for NB and WB telephone systems, of which just a small subset shall be presented in the following. They can be grouped by the type of network that they were developed for, either the Global System for Mobile Communications (GSM) or the code-division multiple access (CDMA) network. All NB and WB codecs with their possible bitrates are listed in tables 3.1 and 3.2, respectively.

In the CDMA network, the enhanced variable rate codec (EVRC) was first published as a NB codec with 3 different bitrates in its first version (EVRC [TIA99]) and with four bitrates in an extended version (EVRC-B [TIA06], see table 3.1). The WB version was published later under the name EVRC-C or EVRC-WB [TIA07] (see table 3.2). It has one mode that operates on WB signals and two modes that operate on NB signals at lower bitrates.

The AMR codec [Jär00] (also called AMR-NB or GSM-AMR) was standardized by European Telecommunications Standards Institute (ETSI) for the NB connections in the GSM network. The WB version of this codec was standardized in 2003 under the name AMR-WB [Bes+02; ITU03].

EVRC		EVRC-B		AMR	
mode	bitrate	mode	bitrate	mode	bitrate
eighth rate	0.80	eighth rate	0.80	1	4.75
		quarter rate	2.00	2	5.15
half rate	4.00	half rate	4.00	3	5.90
full rate	8.55	full rate	8.55	4	6.70
				5	7.40
				6	7.95
				7	10.20
				8	12.20

Table 3.1: Modes of the NB codecs. All bitrates are given in kbps.

EVRC-C		AMR-WB	
mode	bitrate	mode	bitrate
eighth rate	0.80	1	6.60
half rate	4.00	2	8.85
full rate	8.55	3	12.65
		4	14.25
		5	15.85
		6	18.25
		7	19.85
		8	23.05
		9	23.85

Table 3.2: Bitrates of WB codecs. All bitrates are given in kbps.

Chapter 4

Model-Based Artificial Bandwidth Extension System

The aim of ABE is to extend the acoustical bandwidth of speech signals. To achieve this, the missing parts of the frequency spectrum have to be predicted. The prediction can be simplified when the energy distribution and the fine structure of the signal are predicted separately. To separate these parts, the well known source-filter model of speech production is used in the ABE algorithm, which is explained in section 3.2. The separation process with its realization in this work is described in section 4.1.

The aim of ABE is to improve the quality of NB telephone speech (300 Hz–3.4 kHz) by artificially extending the signal to WB (50 Hz–7 kHz). To achieve this, an extension to lower and to higher frequencies is necessary. For the extension to lower frequencies, there are already methods that yield a good signal quality [IMS08]. Therefore, just the extension towards higher frequencies is covered in this thesis. This extension is achieved by estimating the missing UB between 3.4 and 7 kHz based on the NB spectrum. This estimation has to be done accurately because all audible artifacts in the generated speech signal can degrade the speech quality. Most ABE algorithms are based on the source-filter model of speech production explained in section 3.2 [CH94; AHW95; NTN97; EK99; FHG01; JV03b; QK03; HKA05; VZY06; Abe+16]. This implies that the NB speech signal is decomposed into its spectral envelope and the excitation signal, which was in the scope of this thesis implemented according to section 4.1. The advantage of the decomposition is that envelope and excitation can be extended separately in two easier tasks. The extension of the NB excitation can be performed cost-efficiently while the perceived speech quality is at most slightly degraded compared to the WB excitation. This shows that the main difficulty of ABE lies in predicting a good UB spectral envelope based on the NB spectrum. However, there are also recent approaches in which the entire magnitude spectrum is estimated [LL15] or even the complete TD waveform [Lin+18].

Generally, ABE can be implemented in the TD or in the FD. In this thesis, only the FD approach is covered because it allows for a higher flexibility regarding the features that are extracted from the NB signal and it can be combined with other FD-based

algorithms like noise reduction. Another reason is the sensitivity to noise of DNN input features that is discussed in more detail in section 6.2.

A general ABE algorithm in the FD is shown as a block diagram in fig. 4.1. First, the NB signal $s(n)$ is transformed to the FD in an STFT. The NB spectral envelope $S_{\mathrm{nb}}^{\mathrm{env}}(k,l)$ of the speech signal is extracted from the NB spectrum $S_{\mathrm{nb}}(k,l)$ according to section 4.1. The excitation spectrum $S_{\mathrm{nb}}^{\mathrm{exc}}(k,l)$ is then calculated by dividing the NB spectrum by the extracted envelope (see eq. (4.3)).

Both parts are extended to WB spectra separately. Some basic methods for the extension of the excitation spectrum are described in section 4.2. Novel enhancements are presented in chapter 5, based on a related work [Sau+18a]. The extension of the spectral envelope is much more difficult and a variety of different approaches was developed over the last 40 years. The approach of applying a regression DNN is explained in section 4.3. The block called *Envelope Extension* in fig. 4.1 receives the complex NB spectrum $S_{\mathrm{nb}}(k,l)$ as input instead of the NB spectral envelope $S_{\mathrm{nb}}^{\mathrm{env}}(k,l)$ in order to be able to calculate features for the envelope prediction that depend on the complex spectrum. Further enhancements for a DNN-based prediction of the spectral envelope, which have been presented first in two related papers [Sau+18b; Sau+19], are described in chapter 6.

The predicted WB spectrum $\hat{S}_{\mathrm{wb}}(k,l)$ is obtained by multiplying the predicted WB envelope $\hat{S}_{\mathrm{wb}}^{\mathrm{env}}(k,l)$ with the predicted WB excitation $\hat{S}_{\mathrm{wb}}^{\mathrm{exc}}(k,l)$. As $\hat{S}_{\mathrm{wb}}(k,l)$ could contain avoidable artifacts in the NB part, this part is replaced by the original NB spectrum $S_{\mathrm{nb}}(k,l)$. This is achieved by applying a highpass (HP) to the predicted spectrum $\hat{S}_{\mathrm{wb}}(k,l)$ and a lowpass (LP) to the original NB spectrum $S_{\mathrm{nb}}(k,l)$. The filtered spectra are then added up to the final spectrum $\tilde{S}_{\mathrm{wb}}(k,l)$ so that a cross-fading over frequency between the original and the predicted spectrum is realized. Note that the frequency response of the HP filter is obtained by subtracting the zero-phase frequency response of the LP filter from a vector of ones in order to approximately maintain the signal energy. An inverse short-term Fourier transform (ISTFT) that is following the principle of overlap-add is applied to transform the spectrum $\tilde{S}_{\mathrm{wb}}(k,l)$ back to the TD.

4.1 Separation of Envelope and Excitation

Based on the model of speech production [Fan60], a speech signal can be decomposed into two parts: the filter, which is the spectral envelope, and the excitation signal, which is the sound source. An example of the decomposition of a WB spectrum is shown in fig. 4.2. The spectral envelope can e.g. be obtained by LPC analysis or by spectral smoothing of the frequency spectrum. As the ABE algorithm is implemented in the FD, only the smoothing of the spectrum is explained in the following. The smoothed magnitude of a short-term spectrum $S(k,l)$ directly yields the spectral envelope. In this work, the smoothing is realized by a convolution of the spectrum with a moving

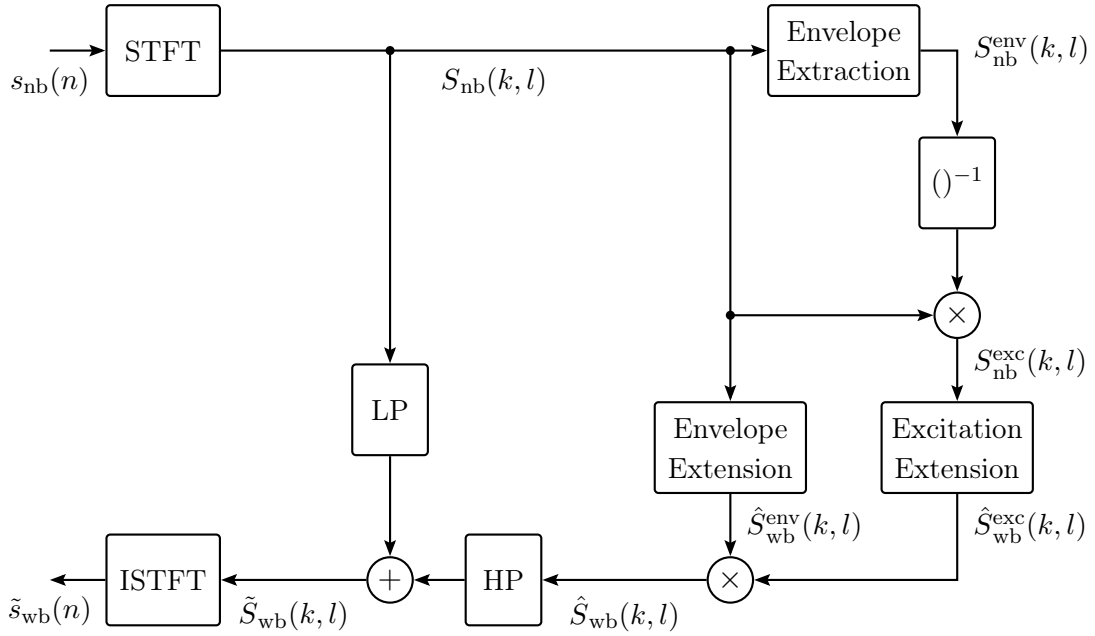
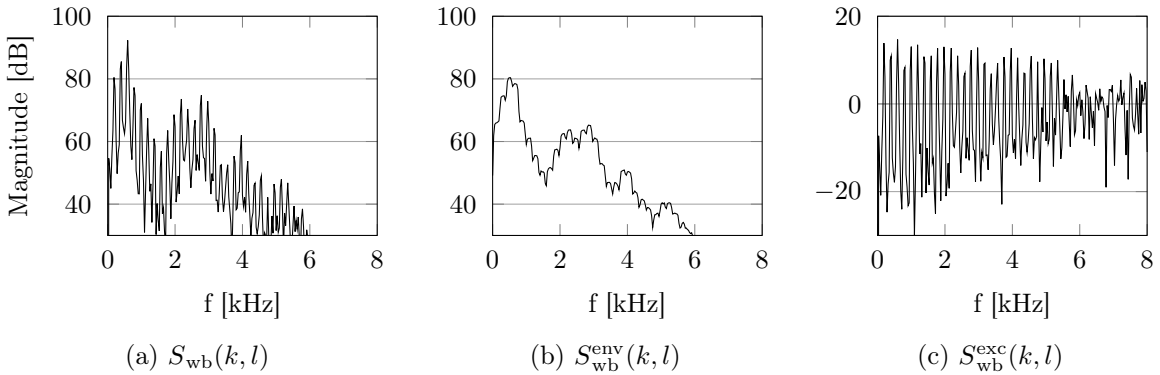


Figure 4.1: General block diagram of ABE in the FD, based on the source-filter model of speech production. Here, LP denotes a lowpass at 3.5 kHz, HP denotes a complementary highpass, STFT denotes the short-term Fourier transform and ISTFT the inverse STFT. The division with the nominator $S_{\mathrm{nb}}(k,l)$ and the denominator $S_{\mathrm{nb}}^{\mathrm{env}}(k,l)$ is realized as a multiplication with the inverse denominator, which is calculated in the block with the exponent -1. Note that the NB signal $s_{\mathrm{nb}}(n)$ and all other signals have a sample rate of 16 kHz. In a real application, an upsampling by the factor of 2 has to be applied to the 8 kHz input signal in order to obtain $s_{\mathrm{nb}}(n)$.



(a) $S_{\mathrm{wb}}(k,l)$ (b) $S_{\mathrm{wb}}^{\mathrm{env}}(k,l)$ (c) $S_{\mathrm{wb}}^{\mathrm{exc}}(k,l)$

Figure 4.2: Spectrum of the vowel [æ] in the word 'favor'. The original WB spectrum $S_{\mathrm{wb}}(k,l)$ (a) is decomposed into the spectral envelope $S_{\mathrm{wb}}^{\mathrm{env}}(k,l)$ (b) and the excitation spectrum $S_{\mathrm{wb}}^{\mathrm{exc}}(k,l)$ (c).

average filter:

$$S^{\text{env}}(k,l) = \frac{1}{k_2(k) - k_1(k)} \sum_{k_0=k_1(k)}^{k_2(k)-1} |S(k_0,l)|\,, \tag{4.1}$$

with the frequency indices

$$\begin{aligned} k_1(k) &= \max\{k - N_{\text{av}}, 0\} \\ k_2(k) &= \min\{k + N_{\text{av}}, K\}\,. \end{aligned} \tag{4.2}$$

The spectral envelope represents the filter in the source-filter model of speech production described in section 3.2. The excitation or fine structure is the quotient of the speech spectrum and its spectral envelope:

$$S^{\text{exc}}(k,l) = \frac{S(k,l)}{S^{\text{env}}(k,l)}. \tag{4.3}$$

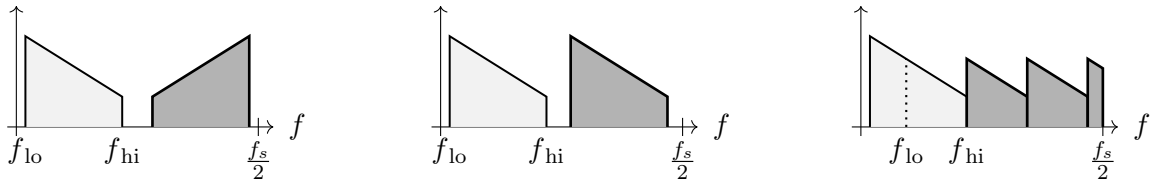
The excitation signal contains all the information about the source in the source-filter model. It can be easily distinguished between a harmonic excitation for voiced speech and pure noise as excitation for unvoiced speech.

4.2 Extension of the Excitation

Most excitation extension methods can broadly be categorized into the following groups: spectral duplication approaches[1], non-linear characteristics, and function generators [IMS08]. The group of spectral duplication approaches contains the methods of spectral folding (SF) (also called *spectral mirroring*), and spectral shifting (SS) (also called *spectral translation*) [MB79]. In many ABE approaches, the original methods of SF or SS are used [CH94; AHW95; EK99; FHG01; Abe+16]. Although other methods for extending the NB excitation were developed, like bandpass-envelope modulated Gaussian noise (BP-MGN) [QK03; HKA05] and the harmonic noise model (HNM) [VZY06], none of them was used as often as the spectral duplication approaches. In the following, only the group of spectral duplication approaches will be further investigated.

The short-term spectrum $S_{\text{nb}}^{\text{exc}}(k,l)$ of the NB excitation signal is calculated according to eq. (4.3) as quotient of the NB spectrum $S_{\text{nb}}(k,l)$ and the corresponding spectral envelope $S_{\text{nb}}^{\text{env}}(k,l)$. In this section, the basic methods of SF and SS are described. A schematic view of how the excitation is shifted in the spectrum is given in fig. 4.3.

[1] This group is called spectral shifting approaches in the cited source. The term spectral duplication is used in order to avoid confusion between the method of spectral shifting and the group of spectral shifting methods.



(a) SF (b) SS (c) MSS

Figure 4.3: Schematic spectra for the three excitation extension methods: SF (a), SS (b), and MSS (c). The dark parts represent the generated extension spectrum, the light parts represent the original NB spectrum.

Spectral Folding The excitation extension method of SF can be implemented efficiently in TD. To mirror the content of 0–4 kHz to 4–8 kHz, a frequency shift of 4 kHz, which is half the sampling rate, has to be applied. This moves the aliasing components from the negative frequencies to the desired region in the UB (see fig. 4.3a). A shift in the FD can be implemented as modulation in the TD with the modulation frequency ω_m:

$$\begin{aligned} \hat{s}_{\mathrm{sf}}^{\mathrm{exc}}(n) &= s_{\mathrm{nb}}^{\mathrm{exc}}(n) + s_{\mathrm{nb}}^{\mathrm{exc}}(n) \cdot \Re\{\mathrm{e}^{\mathrm{j}\omega_m n}\} \\ &= s_{\mathrm{nb}}^{\mathrm{exc}}(n) \cdot (1 + \cos(\omega_m n))\,. \end{aligned} \tag{4.4}$$

For spectral folding, the frequency is set to $\omega_m = \pi$ and the equation can be simplified:

$$\begin{aligned} \hat{s}_{\mathrm{sf}}^{\mathrm{exc}}(n) &= s_{\mathrm{nb}}^{\mathrm{exc}}(n) \cdot (1 + \cos(\pi n)) \\ &= \begin{cases} 2 \cdot s_{\mathrm{nb}}^{\mathrm{exc}}(n) & \text{for } n \text{ even} \\ 0 & \text{for } n \text{ odd} \end{cases}. \end{aligned} \tag{4.5}$$

Like all the other approaches, SF can also be performed in the FD. This is of higher interest in this thesis, because the signal decomposition and the envelope extension are also implemented in the FD. SF was named like this because the NB part of the spectrum with all frequencies smaller than 4 kHz are mirrored (or *folded*) up to frequencies higher than 4 kHz. Additionally, in order to preserve phase consistency, the mirrored values have to be complex conjugated. Subsequently, the estimated WB excitation consists of the NB part and the estimated part:

$$\hat{S}_{\mathrm{sf}}^{\mathrm{exc}}(k,l) = \begin{cases} S_{\mathrm{nb}}^{\mathrm{exc}}(k,l) & \text{for } k < \frac{K-1}{2} \\ S_{\mathrm{nb}}^{\mathrm{exc}*}(K-1-k,l) & \text{for } k \geq \frac{K-1}{2} \end{cases} \qquad \forall k \in \{0,1,\ldots,K-1\}\,. \tag{4.6}$$

Here, S^* denotes the complex conjugate of S, k is the frequency bin, $K = {}^{N}/_{2} + 1$ is the number of frequency bins and N denotes the length of the fast Fourier transform (FFT).

Spectral Shifting SS in its basic form copies the lower half of the spectrum and shifts it to the upper half. This results in a shifting frequency of $f_m = 4\,\mathrm{kHz}$ (see

fig. 4.3b). Again, the modulation could also be applied in the TD. The corresponding modulation frequency in the basic form is $\omega_m = {}^{2\pi f_m}/_{f_s} = {}^{\pi}/_{2}$. The aliasing terms at frequencies below zero are shifted to the NB frequency range and have to be removed in order to not interfere with the NB excitation. This is achieved by a convolution of the shifted signal with a highpass filter h_{hp} that has a cutoff frequency of $f_{\mathrm{c}} = 4\,\mathrm{kHz}$. The predicted WB excitation can then be formulated as

$$\hat{s}_{\mathrm{ss}}^{\mathrm{exc}}(n) = s_{\mathrm{nb}}^{\mathrm{exc}}(n) + s_{\mathrm{nb}}^{\mathrm{exc}}(n) \cdot \cos\left(n \cdot \frac{\pi}{2}\right) * h_{\mathrm{hp}}\,. \tag{4.7}$$

In the corresponding FD implementation, the NB spectrum is copied to the upper frequency band ($f > 4\,\mathrm{kHz}$ or $k > {}^{K-1}/_{2}$):

$$\hat{S}_{\mathrm{ss}}^{\mathrm{exc}}(k,l) = \begin{cases} S_{\mathrm{nb}}^{\mathrm{exc}}(k,l) & \text{for } k < \frac{K-1}{2} \\ S_{\mathrm{nb}}^{\mathrm{exc}}(k - \frac{K-1}{2}, l) & \text{for } k \geq \frac{K-1}{2} \end{cases} \qquad \forall k \in \{0, 1, \ldots, K-1\}\,. \tag{4.8}$$

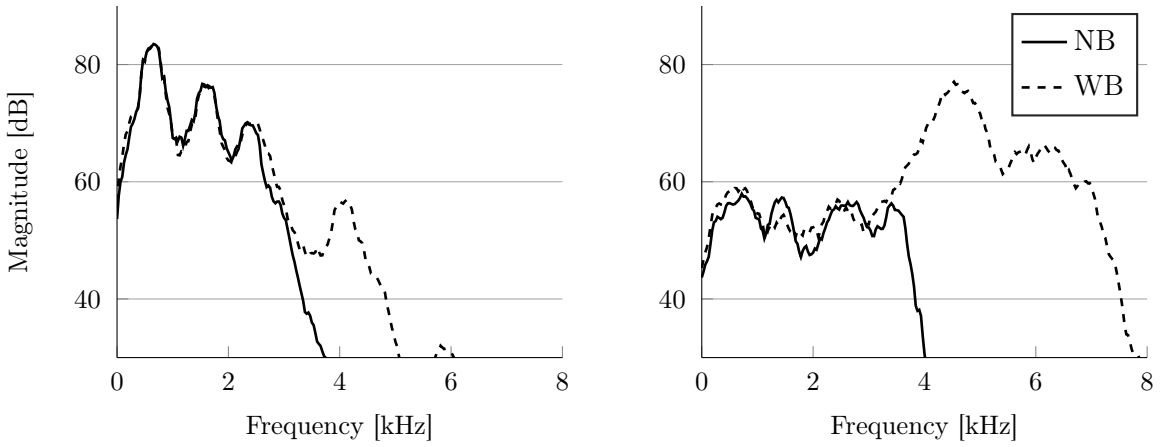
4.3 Extension of the Spectral Envelope

While the differences between the original and the estimated UB excitation are only slightly audible, the error in estimating the spectral envelope is the main factor that degrades the quality of the predicted WB speech. A comparison between six ABE approaches in 2016 showed that the speech quality gap between transcoded NB and transcoded WB speech could still not be closed to 50 % [Abe+16]. Dependent on the language, the results are even worse. For German sentences, e.g., none of the six compared ABE methods yielded a closing of the gap of more than 25 % [Abe+16]. This shows the difficulty of this task and that the perceived speech quality is highly sensitive to the UB spectral envelope.

All approaches of UB envelope estimation map a representation of a NB spectrum or envelope to a representation of an UB or WB envelope. In fig. 4.4, exemplary pairs of NB and WB short-term spectral envelopes are shown for a vowel and a fricative. The prediction of the UB or WB envelope can be achieved in a classification task, e.g. by using a codebook, or in a regression task by estimating the energy directly. Because of the success of regression DNNs for ABE in the last 10 years [AF18; Li+18; LK16; GL15; PA11], this approach is further investigated.

As input for the DNN, a feature vector $\mathbf{x}(l)$ has to be extracted from the NB spectral envelope $S_{\mathrm{nb}}(k,l)$ for each frame. It shall contain useful information for the DNN related to the prediction task. Usually, a representation of the NB spectrum or its spectral envelope is contained in the feature vector. Different representations like mel spectra and MFCCs are explained in section 3.4. A feature selection method can help to find a set of useful input features. In section 6.2, this topic will be further investigated. Based on the input feature vector $\mathbf{x}(l)$, the network estimates the output feature vector[2] $\hat{\mathbf{y}}(l,\Theta)$.

[2]The dependency of Θ for all predicted values will be omitted in the following to achieve a better readability.



(a) [æ] in that (b) [s] in ask

Figure 4.4: WB (dashed) and NB (solid) short-term spectral envelopes of the phoneme [æ] in the word 'that' (a) and the phoneme [s] in the word 'ask' (b). The spectra were extracted from recordings that were made with a network simulator and a cellphone. The NB signal was recorded after the simulated transmission with the AMR codec at 12.2 kbps and the WB signal after the AMR-WB codec at 12.65 kbps.

These output features are processed in the feature synthesis in order to obtain the WB spectral envelope $\hat{S}_{\text{wb}}^{\text{env}}(k, l)$. The output feature is usually defined as a representation of the WB spectral envelope. Sometimes, additional parameters are predicted that are used in the synthesis step. The whole process of obtaining the WB spectral envelope from the NB spectral envelope is depicted in fig. 4.5

Besides different DNN input and target features, various DNN structures have been employed. Basic MLPs, feedforward neural networks with fully connected layers, were used first [IS03; KLA07; PA11]. These networks predict the output vector based on the current input vector. This means that no dependencies to the last frames can be used although there are dependencies in human speech signals. A workaround is to feed either delta features to the DNN or to concatenate a series of input feature vectors so that all information of multiple time-steps can be used. In order to model the time dependencies more accurately, several types of RNNs have been applied to ABE [Wan+16; GLD16]. However, this does not change the way how the network is integrated into the ABE setup.

The DNN that shall predict the WB or UB envelope has to be trained on data before it can be used for ABE. Pairs of NB and WB speech signals have to be generated in order to extract the representations that are chosen in the input and output features. A WB signal is sufficient to generate a 16 kHz NB signal by a convolution with a BP filter that represents the NB characteristics. Further methods of simulating the NB input signal are discussed in section 6.1. The generated data has to be normalized to zero mean and unit variance before it is passed to the DNN by subtracting the mean and dividing the result by the standard deviation (see fig. 4.5). The mean and the

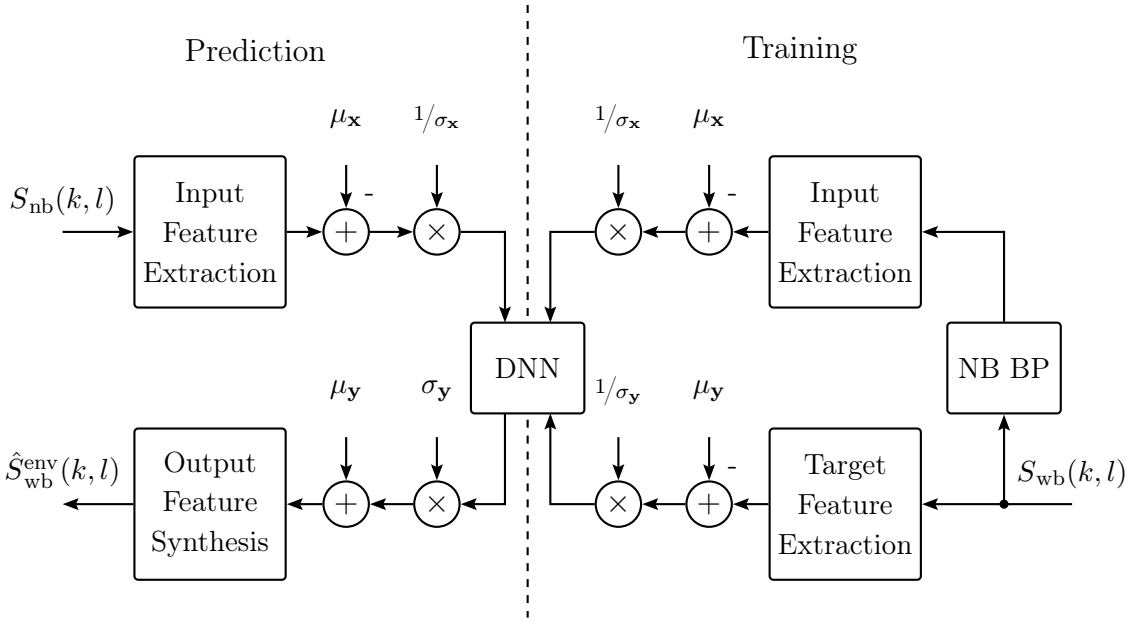


Figure 4.5: Block diagram of the training of a DNN (right part) and the prediction of a WB envelope using the trained DNN (left part). The prediction part of this diagram corresponds to the block *Envelope Extension* in fig. 4.1. The DNN in the middle is either connected to the left or to the right side. The mean values $\mu_{\mathbf{x}}$ and $\mu_{\mathbf{y}}$ and the standard deviations $\sigma_{\mathbf{x}}$ and $\sigma_{\mathbf{y}}$ are calculated based on the input and target data of the whole training dataset, respectively. The training part receives just a WB spectrum as input that can be obtained by an STFT applied to a speech signal.

standard deviation, which are calculated on the whole dataset, are stored in order to be able to normalize the input features and de-normalize the predicted output features later when the DNN is integrated into the application.

4.4 Synthesis of the Extended Signal

The product of the extended excitation and the extended spectral envelope yields an estimation of the UB or WB spectrum, depending on the implementation. When the WB spectrum is estimated, the NB part of the spectrum is estimated although it is already known. The aim is to keep the original NB signal unchanged and just replace the missing parts with the estimation. The simplest method is to combine the NB spectrum up to 3.4 kHz with the WB spectrum starting at 3.4 kHz. As the cellphone manufacturers can choose a cutoff filter for the NB sending characteristics freely in a defined range, the cutoff frequency differs between 3.4 and 3.9 kHz. In order to yield a smooth transition between original NB and estimated UB, cross-fading can be applied to the two spectra in FD. A cross-fading at 3.4 kHz could overwrite already known values between 3.4 and 4 kHz by the prediction. However, a cross-fading at 4 kHz would insert a spectral gap when the cutoff frequency of the cellphone is low. This problem can be at least alleviated by combining a fade-in of the UB spectrum at 3.4 kHz with a maximum operation that assures that no spectral gap can occur above this frequency when the cutoff frequency of the cellphone is low. By taking the maximum of the complete NB signal and the highpass filtered UB signal in every bin, the estimation can only increase the energy of the NB spectrum. The cross-fading is realized in fig. 4.1 as combination of a HP filter and a complementary LP filter.

Chapter 5

Enhanced Extension of the Excitation Signal

This chapter deals with problems and advances regarding the extension of the excitation signal and is mainly based on the results of a prior publication [Sau+18a]. Subjective listening tests showed that there are some audible artifacts introduced by the excitation extension from NB to WB using spectral duplication methods [MB79]. This is proved by the listening test results presented in section 7.4. Regarding the obvious differences in the spectrum, the following three drawbacks of classical excitation extension methods can be noticed:

1. **Spectral gap:** The extended spectrum is interrupted by a spectral gap around 4 kHz. This is caused by the cutoff frequency of the NB characteristics that is mostly located between 3.4 and 3.9 kHz. The excitation between this cutoff frequency and 4 kHz is missing for SS. For SF, this gap is doubled because of the mirroring of the spectrum at 4 kHz.

2. **Shifted harmonics:** In a voiced spectrum, the harmonics occur at multiples of the fundamental frequency. The spectral components that are copied from the NB spectrum are not inserted at multiples of the pitch.

3. **Pitch change inconsistency:** A pitch change over time is not scaled up to higher harmonics properly. In a voiced spectrogram in which the pitch rises from one frame to another, the frequency difference between these frames is scaled by the index of each harmonic. This means that the delta frequency is twice as big for the first harmonic, three times for the second and so on. In other words, the slope of the curve described by each harmonic is scaled with the index of the harmonic.

Problem 1 occurs especially for SF, where the spectral gap is twice the difference between 4 kHz and the cutoff frequency of the NB characteristics, and for SS with a fixed shift of 4 kHz, where the spectral gap is only once the described difference. It could be solved by choosing SS with a frequency shift of 3.4 kHz. However, spectral

gaps of a small size were mostly inaudible in subjective listening tests in [JV03b]. The other problems will be in the focus in the following. Problems 2 and 3 could be solved by pitch adaptive modulation. For this method, in a first step, the voiced frames have to be detected as the modulation is just applied to those. In a second step, the pitch of the voiced frames is estimated. The spectral shift is then adapted to a multiple of the pitch frequency for each voiced frame. It was implemented several times [FHG01; Kor01; JV03b] but there are two main drawbacks: it only works well with a robust pitch estimation and voiced/unvoiced detection and the pitch estimator has a relatively high computational cost. Additionally, the errors introduced by inserting the harmonics at a wrong position in the spectrum did not affect the perceived speech quality significantly in [FHG01; JV03b]. Altogether, the improvement in the extension of the excitation that comes with pitch-adaptive filtering is only marginal [IMS08]. Finally, for a robust and cost-efficient algorithm, shifting with a fixed frequency seems more suitable. Problem 3 is related to problem 2 and can occur in different strengths: In SF, a change of the fundamental frequency over time introduces a change of the harmonics in the extended UB in the opposite direction. This can cause a 'somewhat garbled sound' [FHG01], especially for high pitch variations. This problem might be reduced by SS, where the change of the UB harmonics has at least the right direction, although the slope is still too low.

As the potential for subjectively perceived improvements regarding problems 1 to 3 seem to be rather small, other differences between the extended and the original WB excitation might cause the existing quality gap and are therefore investigated in the following. Two further problems of SF and SS are addressed in [Sau+18a]:

4. **Disturbingly flat harmonics in the UB:** This problem is a special case of problem 3. In a voiced speech spectrogram, the change of the n_h-th harmonic between two adjacent time frames f_{Δ,n_h} is $n_\mathrm{h}+1$ times the change of the fundamental frequency: $f_{\Delta,n_\mathrm{h}} = (n_\mathrm{h}+1) \cdot f_{\Delta,0}$. This means that the curves that are described by each harmonic are less flat for rising n_h. Consequently, with SS, the curve that belongs to the first inserted UB harmonics at around 4 kHz is too flat. The variation of the harmonics is rather high normally in this frequency range, mostly more than 15 times higher than the variation of the fundamental frequency. These flat harmonics introduce a whistling noise that can degrade the perceived speech quality [Sau+18a]. The effect is perceived even stronger when the pitch frequency varies a lot, like it does between 0.2 and 0.7 seconds in fig. 5.1d. With SF, a similar effect occurs with the highest inserted harmonics directly below 8 kHz (see fig. 5.1c). The problem can only be solved in case the pitch and the first harmonics with a flat curve over time are not copied to the UB.

5. **Constant harmonicity:** Another difference that can be observed in the spectrogram is a mismatch in the harmonics-to-noise ratio (HNR), which is the ratio between the magnitude of a harmonic and the magnitude between two harmon-

ics [Sau+18a]. The harmonic structure of the excitation in voiced phonemes generally gets less dominant towards higher frequencies, which means that the HNR decreases. An example of an excitation spectrum is shown in fig. 4.2c. The excitation extension methods that copy the whole NB excitation to the UB do not take this effect into account, because the HNR in the UB is also just a copy from the NB HNR. As a consequence, the HNR is too high in the extended part of the spectrum. This might also lead to audible artifacts.

In a prior work [Sau+18a], a new variant of SS that was named multiple spectral shifting (MSS) was introduced to overcome problem 4. It is based on multiple spectral shifts of a smaller frequency range of the NB excitation that does not include the fundamental frequency.

Problem 5 addresses the deviation of the harmonicity in high frequencies. The resulting HNR in the UB after excitation extension is equal to the HNR in the NB where it was copied from. Therefore, it was suggested to add comfort noise in higher frequencies to manually decrease the HNR. The MSS approach with this extension was called multiple spectral shifting with comfort noise (MSSCN) in [Sau+18a]. An example excitation spectrogram is shown for original WB speech, NB speech, and the extended versions using SF, SS, MSS, and MSSCN in fig. 5.1.

The suggested method of MSS is presented in section 5.1. The extension with comfort noise and its combination with MSS is shown in section 5.2. The performance of the proposed algorithms is evaluated in section 7.4.

5.1 Multiple Spectral Shifting

In order to solve problem 4, multiple spectral shifting (MSS) was suggested in [Sau+18a]. Problem 4 is caused by copying the lower part of the NB excitation spectrum to the UB. MSS does not reuse this part of the spectrum for the extension, where the shape of the harmonics over time is rather flat. A fixed threshold frequency f_{lo} is set as lower limit of the frequency range that shall be copied. Additionally, the negative effects described in problem 5 might be reduced with MSS. The low-frequency part of the excitation is the part with the highest HNR, which might introduce artifacts when it is copied to higher frequencies. After informal listening tests, the frequency range $[f_{\mathrm{lo}}, f_{\mathrm{hi}}]$ that is copied to the UB was set to $f_{\mathrm{lo}} = 1.5\,\mathrm{kHz}$ and $f_{\mathrm{hi}} = 3.5\,\mathrm{kHz}$ [Sau+18a]. This part of the spectrum is copied and shifted multiple times until the bandwidth of the WB spectrum is reached (see fig. 4.3c). The modulation frequency ω_m that corresponds to the frequency range $f_m = f_{\mathrm{hi}} - f_{\mathrm{lo}} = 2\,\mathrm{kHz}$ has to be set to

$$\omega_m = \frac{2\pi f_m}{f_s} = \frac{\pi}{4}. \tag{5.1}$$

In a TD implementation, the BP-filtered signal is only modulated twice because the maximum frequency of the extension of 7.5 kHz already exceeds the 7 kHz cutoff of

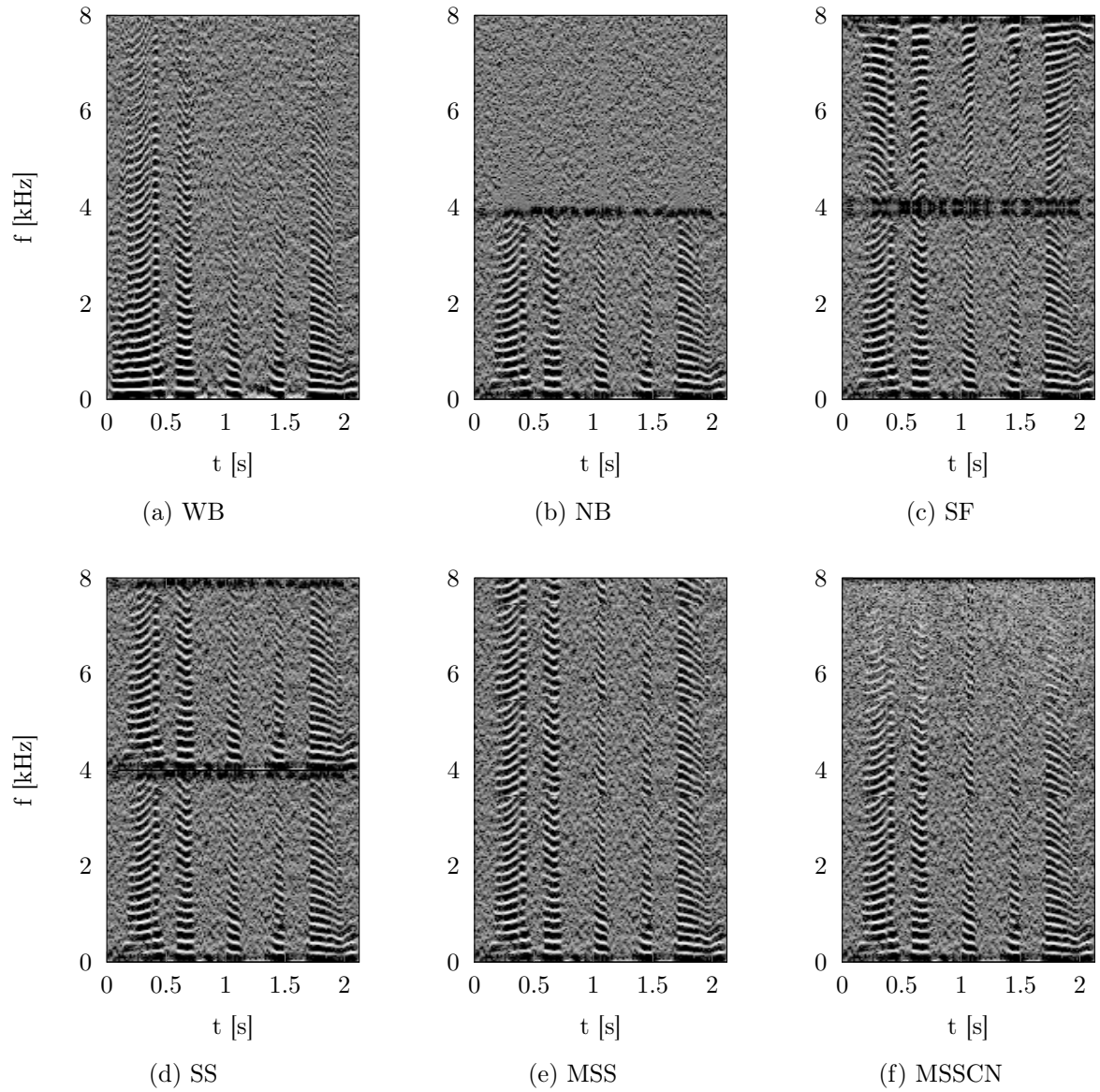


Figure 5.1: Spectrograms of a speech excitation signal. Higher magnitudes are represented by brighter shades. The original WB excitation $S_{\mathrm{wb}}^{\mathrm{exc}}(k,l)$ (a) and the NB excitation $S_{\mathrm{nb}}^{\mathrm{exc}}(k,l)$ (b) are compared to four extended excitations. All four extensions are based on the NB excitation spectrum: SF ($\hat{S}_{\mathrm{sf}}^{\mathrm{exc}}(k,l)$) in (c), SS ($\hat{S}_{\mathrm{ss}}^{\mathrm{exc}}(k,l)$) in (d), MSS ($\hat{S}_{\mathrm{mss}}^{\mathrm{exc}}(k,l)$) in (e), and MSSCN ($\hat{S}_{\mathrm{msscn}}^{\mathrm{exc}}(k,l)$) in (f). The frequency at which the insertion of the UB excitation starts is 4 kHz in (c) and (d) and 3.5 kHz in (e) and (f).

WB codecs and the third modulation would insert undesired aliasing components if no further filtering was applied:

$$\hat{s}^{\text{exc}}_{\text{mss}}(n) = s^{\text{exc}}_{\text{nb}}(n) * h_{\text{lp}} + (s^{\text{exc}}_{\text{nb}}(n) * h_{\text{bp}}) \cdot \left(\cos\left(n\frac{\pi}{4}\right) + \cos\left(2n\frac{\pi}{4}\right)\right) * h_{\text{hp}}, \tag{5.2}$$

where h_{lp} denotes a LP filter with the cutoff frequency f_{lo}, h_{bp} denotes a BP filter for the frequency range $[f_{\text{lo}}, f_{\text{hi}}]$, and h_{hp} denotes a HP filter with the cutoff frequency f_{hi}. The corresponding FD implementation with the discrete frequency range $[k_{\text{lo}}, k_{\text{hi}}]$ extends the excitation up to the Nyquist frequency:

$$\hat{S}^{\text{exc}}_{\text{mss}}(k,l) = \begin{cases} S^{\text{exc}}_{\text{nb}}(k,l) & \text{for } k < k_{\text{hi}} \\ S^{\text{exc}}_{\text{nb}}(k_{\text{lo}} + ((k-k_{\text{hi}}) \bmod k_{\Delta}), l) & \text{for } k \geq k_{\text{hi}} \end{cases}, \tag{5.3}$$

with $k_{\Delta} = k_{\text{hi}} - k_{\text{lo}}$. The FD implementation will be used in the evaluation of this thesis in chapter 7.

5.2 MSS with Comfort Noise

The approach of MSS can mainly solve problem 4 but problem 5 still persists, although in a slightly alleviated form. In the related publication to this chapter [Sau+18a], an extension to MSS was proposed that addressed the issue of the constant HNR by inserting comfort noise. Consequently, this new approach was named multiple spectral shifting with comfort noise (MSSCN). The extended excitation signal is interpolated with a spectrum of white noise $Z(k)$ in order to reduce the HNR towards higher frequencies. The effect can be seen by comparing fig. 5.1e and fig. 5.1f. A frequency dependent weighting factor $\beta(k)$ is defined for the interpolation. The real and the imaginary part of the complex noise spectrum $Z(k)$ follow a normal distribution with zero mean and unit variance. The mean energy of the excitation is systematically normalized to roughly about 0 dB (see fig. 4.2c). As the NB part should stay unchanged, comfort noise is inserted for $k > k_{\text{hi}}$ only [Sau+18a]:

$$\hat{S}^{\text{exc}}_{\text{msscn}}(k) = \sqrt{1-\beta^2(k)} \cdot \hat{S}^{\text{exc}}_{\text{mss}}(k) + \beta(k) \cdot Z(k), \tag{5.4}$$

where the weight factor β is calculated according to:

$$\beta(k) = \begin{cases} 0 & \text{for } k < k_{\text{hi}} \\ \left(\frac{k-k_{\text{hi}}}{K-1-k_{\text{hi}}}\right)^{\rho} & \text{for } k \geq k_{\text{hi}} \end{cases}. \tag{5.5}$$

The exponent ρ was set to 0.7 after informal listening tests. A listening test that compares the proposed excitation methods with the classical methods is described in chapter 7.

Chapter 6

Improved DNN Training for Spectral Envelope Extension

This chapter comprises novel approaches for the training of a DNN that shall predict a WB spectral envelope from a NB spectrum in an ABE application. These approaches aim at various parts of the training, like the generation process of the training data, the selection of the input and output features of the DNN, and different loss functions and architectures. The basic structure of the ABE method stays the same as described in chapter 4 for all of these enhancements.

Suitable training data is one of the most important requirements for a successful DNN training that shall be robust against variations of the input data. In this case, suitable refers to training data that contains most of the variations that might occur in reality in the prediction stage. This helps the network to generalize well as it cannot rely on a special characteristics of the input data. In order to achieve this, the training data is modified in many dimensions following the principle of multi-condition training (see section section 2.2.7). The whole process of data generation is explained in section 6.1.

Another important task for the training of a DNN is the definition of its inputs and outputs. Section 6.2 deals with the selection of input and target features for a regression DNN that shall predict the WB envelope in an ABE application. For the target feature, a compact representation of a WB envelope is chosen. For the selection of the input features, a pool of features is defined and the forward selection method is explained in detail. The results and the evaluation are given in section 7.3.

The complete training setup for the prediction of a WB spectral envelope that was used in the scope of this thesis is given in section 6.3. The focus lies on the general methods and parameters that are used equally for all of the trainings. Some parameters that were chosen differently for the evaluations are explained separately in the respective sections.

Advanced methods of training the DNN without only relying on the MSE as a loss function are presented in the last two sections. Discriminative training is proposed in section 6.4, following the prior work in [Sau+18b]. It shall make use of the differences between phoneme classes to avoid a lisping effect in the prediction.

The proposed method in section 6.5 is adversarial training for ABE envelope estimation. This section is based on the findings from [Sau+19]. Adversarial training is an alternative for classical DNN training with a standard loss function. Instead of using a loss function to evaluate the error, a second DNN is trained for this task. The backpropagation algorithm (see section 2.2.2) can be used as well, because the evaluating network can be derived completely.

6.1 Data Generation

A key task for DNN training is the selection of suitable training data. In order to train a robust and well performing DNN, the training data set has to fulfill the following conditions:

- **Large amount of data**
  If the amount of training data is too small for a given complexity of the DNN model, overfitting might occur. This means that the network only performs well on the training data and that the performance on different data is not optimal. This effect can be observed by comparing the loss function applied to the training and the validation data. Overfitting occurs when the validation loss increases again while the loss on the training data further decreases.

- **High variation in parameters**
  It is important to cover as many different configurations and properties of the training data as possible. The DNN has to know about the variation that can occur in order to perform robustly. When a characteristics is never shown to the network in the training stage, it might occur that the prediction is not accurate.

- **Balanced classes**
  In case the training dataset can be subdivided into several classes, the amount of data for each of these classes has to be chosen consciously. Data that belongs to a class that was underrepresented in the training might not perform well in the prediction because of the mean operation over all training examples in the loss function. An example can be given regarding different phoneme classes: as sibilant fricatives occur in just roughly 10% of the TIMIT dataset, they are underrepresented compared to other phonemes.

In the field of ABE, a pair of NB and WB speech signals can be generated from every WB speech signal by a convolution with a filter that models the NB characteristics. This enables us to build a large set of training data with low effort, so that the amount of data might not be a limiting factor in this case. An example for the generation of training data is given in section 7.1.2.

The second condition of a high variation can only be fulfilled to a certain extent because there are too many dimensions in which speech data varies. Obvious examples are the speaker (including the speaker's gender) and the speaker's language. Another

area of parameters is given by the recording conditions like the loudness, the impulse response of the microphone, the impulse response of the room, and many more. The background noise conditions can also have an effect, like the signal-to-noise ratio (SNR) and the noise type. For NB telephony, the frequency response is not exactly normed, there are just some frequency dependent limits of the attenuation that should not be exceeded (see section 3.3). The differences in the frequency response can be modeled with a set of slightly varying FIR filters that are used to generate NB speech from WB speech. Finally, a speech signal that was transmitted via a cellular system gets coded and decoded. Different codecs with various configurations can be simulated offline. These are some of the important ways in which the speech signal can vary. Variation in all of these dimensions is necessary for a good training result. The implementation of the data generation is described in section 7.1.2.

The third condition could be fulfilled by adjusting the amount of data that belongs to different classes. First, classes of training examples that are handled differently by the algorithm have to be found. The difference between sibilant fricatives and other phonemes regarding ABE was already shown in section 3.1. Regarding this example, it is important to have enough training examples with sibilant fricatives in the training set. Instead of creating balanced classes, there are also some approaches for the training of unbalanced data. These include a stronger weighting of examples that belong to underrepresented classes and problem-specific changes in the loss function that yield similar results.

6.2 DNN Feature Selection

Before training a DNN, input and target features of the network have to be chosen carefully. In most problems, the input and target data can be directly taken as raw input and target features. However, this is often not efficient as the dimensions could be reduced in a compressed representation. Therefore, the aim is to find a compact feature set that enables the DNN to yield good prediction results. A single compact input feature might lose some parts of the information given in the raw data. Additional specific input features can help the DNN to efficiently predict accurate target values by providing the lost information in a compact form.

Otherwise, if only one input feature was given, the first layer in the network could perform some of the calculations of the feature extraction. This would require higher network dimensions caused by the additional layer. As the matrix multiplication is rather expensive, computation time can be saved by calculating the most relevant input features efficiently and at the same time keeping the network dimensions small. The way of defining and selecting suitable features is described in this chapter following the results of a related paper [SFS18]. As a theoretical basis, some common feature selection algorithms are described in section 2.4.

Selecting the target feature is often an easier task than selecting the input feature. Mostly, one feature is sufficient and the selection is often already given by the task.

In this thesis, the task for the DNN is to map a NB spectral envelope to a WB spectral envelope. Consequently, the WB spectral envelope is the single target feature. Section 6.2.1 deals with choosing a good representation of the WB envelope as DNN output.

For the selection of the input features, a pool of features that can be extracted from the NB speech is necessary. Such a pool of input features for ABE has already been published in [Jax02]. Some of the contained features are explained briefly in section 6.2.2. Additionally, some FD-based features were suggested according to [SFS18] in order to be less sensitive to background noise and to achieve a pure FD-based implementation. Comparisons between the TD features and suitable FD features that can partly replace their functionality with a higher robustness are shown in section 6.2.3 based on example sound signals.

A forward selection approach was applied to ABE based on the feature pool and the selected target feature in order to find a good input feature set. The approach is described in section 6.2.4. A main difficulty in ABE is the correct extension of sibilant fricatives like [s] and [z] [BAF14]. They will be hard to detect in the NB spectrum and the perceived speech quality will be highly degraded if the inserted UB energy is too low [BAF14]. Therefore, it was investigated which input feature sets yield a high accuracy in the detection of sibilant fricatives. This was done by also applying the forward selection method to a classification DNN that detects sibilant fricatives. The results of the selection approach are given in section 7.3.

6.2.1 Target Feature Selection

There are multiple representations for a spectral envelope besides the magnitude spectrum, including linear predictive coding (LPC) coefficients, the mel spectrum and mel frequency cepstral coefficients (MFCCs). As the energy distribution of human speech in the UB is rather smooth, a limited number of coefficients is sufficient to model the UB. This shows that the resolution of the magnitude spectrum is higher than necessary. With the FFT length of 512, which is used in the trainings in this thesis, 128 bins would be needed for the upper half of the spectrum. LPC coefficients represent a spectrum in a very compressed way but they are highly sensitive to small changes. This makes them unusable in a regression task that shall yield a high robustness. Line spectral frequencies (LSFs) can be used instead to model the energy distribution [IMS08]. Alternatives that are calculated from the FD spectrum are the mel spectrum and MFCCs. Informal listening tests revealed that 40 mel filters and 30 MFCCs are sufficient to reconstruct a WB spectral envelope with mostly inaudible deviations. As the whole training is implemented in the FD and as there are no obvious advantages in using LSFs, they are not covered by this thesis. In several informal tests, the performance of ABE systems with 40 mel coefficients and 30 MFCCs was compared. Especially trainings were run with one of those representations as input and output feature. Because of the better performance of MFCCs, 30 MFCCs were chosen as output feature for all the regression

DNNs in this thesis.

6.2.2 Input Features

A set of well selected input features allow for a compact network architecture. With a feature selection method, this set can be found algorithmically. As a basis for the feature selection method, a pool of 10 input features was created, which is shown in table 6.1. It was first presented in [SFS18]. All features are based on the current frame of the buffered NB speech signal. The TD features *local kurtosis*, *gradient index* and *zero crossing rate* have already been used successfully for ABE in [Jax02] and, in combination with DNNs, in [Abe+16; AF17]. Their definitions are adapted from [Jax02]. The FD features *MFCCs*, *spectral centroid* and *signal power level* were also used in slightly different realizations in [Jax02]. In order to increase the feature robustness regarding background noise, the goal is to extract the features directly in the FD after a Wiener-filter based noise suppression. Still, the TD features are highly sensitive to background noise. In order to reduce this sensitivity, the remaining features *onset*, *offset*, *signal above noise* and *high spectral centroid* were proposed in [SFS18]. Δ- and $\Delta\Delta$-features are also regarded for those features that show a good performance. A Δ-feature $\mathbf{x}^{\Delta}(l)$ represents the first derivative of a feature vector $\mathbf{x}(l)$ and is calculated as the difference between two adjacent time frames:

$$\mathbf{x}^{\Delta}(l) = \mathbf{x}(l) - \mathbf{x}(l-1)\,. \tag{6.1}$$

The one-sided calculation of Δ values was chosen because two-sided, symmetric Δ-features would add an additional delay of one frame to the algorithm. This is not desired in ABE which is running in real-time. Applying the equation to Δ-features again leads to $\Delta\Delta$-features:

$$\mathbf{x}^{\Delta\Delta}(l) = \mathbf{x}^{\Delta}(l) - \mathbf{x}^{\Delta}(l-1)\,. \tag{6.2}$$

The following list gives an overview of all features from the pool with its definitions. Every feature is shortly described and a brief motivation is given for the FD features which were proposed in [SFS18]. The NB signal in the TD is named $s_{\mathrm{nb}}(n)$, where n is the sample index. The NB short-term spectrum in the FD is named $S_{\mathrm{nb}}(k,l)$, where k is the frequency index and l is the time frame index. Some features are based on the NB power spectrum (PS), which is the squared absolute NB spectrum (see eq. (3.2)).

Features that were used by Jax [Jax02] in a similar form:

- **MFCC**: Cepstral coefficients define the logarithmic spectral envelope of a speech signal in a condensed form. MFCCs are based on the mel-frequency PS of the NB spectrum $\Phi_{S'S',\mathrm{nb}}(k',l)$ which consists of $K' = 40$ mel bands in this work. The transformation from a mel spectrum to MFCCs and back is explained in

Feature name	Abbreviation	Domain	Dimension
MFCCs	mfc	FD	30/20/10
spectral centroid	cen	FD	1
high spectral centroid	hce	FD	1
signal power level	lev	FD	1
local kurtosis	kur	TD	1
gradient index	gri	TD	1
zero crossing rate	zcr	TD	1
onset probability	ons	FD	4
offset probability	off	FD	4
signal above noise	san	FD	4

Table 6.1: Feature pool for the input of a DNN, consisting of TD and FD features. The first features above the horizontal line are taken from the publication by Jax [Jax02]. The features below the horizontal line were proposed in the related paper [SFS18]. Each time a feature is added to the set, it is removed from the pool and replaced by its Δ-feature. The same holds for Δ-features, which are replaced by ΔΔ-features. For most of the features, the dimension of Δ and ΔΔ-features are the same. Only for the MFCC feature, the dimension is reduced to 20 for the Δ-feature and to 10 for the ΔΔ-feature. This table was first presented in [SFS18].

section 3.4.3 in eqs. (3.9) and (3.10). The logarithmic mel spectrum $\Phi^{[\mathrm{dB}]}_{S'S',\mathrm{nb}}(k',l)$ is transformed to MFCCs following eq. (3.9). The basic input feature vector consists of 30 MFCCs of the NB envelope:

$$x_{\mathrm{mfc}}(i_c,l) = c_{\mathrm{nb}}(i_c,l), \tag{6.3}$$

where i_c is the index of the cepstral coefficient.

- **Spectral centroid**: The spectral centroid is the sum of the spectral magnitude in all frequency bands, where each summand is weighted with the normalized frequency [Jax02]. The resulting relative frequency value gives a hint where the main part of the energy is concentrated in the spectrum. It can be used as a measure for voiced-/unvoiced-detection because voiced frames have a high amount of energy in very low frequencies. The spectral centroid (also called *spectral balance*) is calculated as follows:

$$x_{\mathrm{cen}}(l) = \frac{\sum_{k=0}^{K-1} k \cdot |S_{\mathrm{nb}}(k,l)|}{K \sum_{k=0}^{K-1} |S_{\mathrm{nb}}(k,l)|}. \tag{6.4}$$

- **High spectral centroid**: The spectral centroid mainly differentiates between voiced and unvoiced frames. In order to get more detailed information about the upper part of the NB spectrum, the lower frequencies are excluded from the

calculation of the high spectral centroid. The only difference to the calculation of the centroid is that the frequency range is set to 3–4 kHz. This is achieved by replacing the limits of the frequency index k in eq. (6.4):

$$x_{\mathrm{hce}}(l) = \frac{\sum_{k=3\cdot(K-1)/4}^{K-1} k \cdot |S_{\mathrm{nb}}(k,l)|}{K \sum_{k=3\cdot(K-1)/4}^{K-1} |S_{\mathrm{nb}}(k,l)|}. \tag{6.5}$$

- **Signal power level**: The signal power level is defined as the logarithmic power value of each frame in dB:

$$x_{\mathrm{lev}}(l) = 10 \log_{10} \sum_{k=0}^{K-1} \Phi_{SS,\mathrm{nb}}(k,l). \tag{6.6}$$

It can help to detect speech pauses by only evaluating a single value in case the background noise is low enough.

- **Local kurtosis**: The local kurtosis is calculated in the TD and can be used for the detection of onsets or voicing [JV04]:

$$x_{\mathrm{kur}}(l) = \log_{10} \frac{\frac{1}{N} \sum_{n=lF}^{lF+N-1} (s_{\mathrm{nb}}(n))^4}{\left(\frac{1}{N} \sum_{n=lF}^{lF+N-1} (s_{\mathrm{nb}}(n))^2 \right)^2}. \tag{6.7}$$

- **Gradient index**: The gradient index is a TD measure for voiced-/unvoiced-detection [JV04]:

$$x_{\mathrm{gri}}(l) = \frac{\sum_{n=2+lF}^{lF+N-1} \Psi(n) \, |s_{\mathrm{nb}}(n) - s_{\mathrm{nb}}(n-1)|}{\sqrt{\sum_{n=2+lF}^{lF+N-1} (s_{\mathrm{nb}}(n))^2}}, \tag{6.8}$$

with

$$\Psi(n) = 0.5 \cdot |\psi(n) - \psi(n-1)|, \tag{6.9}$$

and $\psi(n)$ defined as

$$\psi(n) = (s_{\mathrm{nb}}(n) - s_{\mathrm{nb}}(n-1)) / |s_{\mathrm{nb}}(n) - s_{\mathrm{nb}}(n-1)|. \tag{6.10}$$

$\psi(n)$ can be interpreted as the normed gradient of the signal $s_{\mathrm{nb}}(n)$ and Ψ as an indicator for a change in the normed gradient $\psi(n)$.

- **Zero crossing rate**: The zero crossing rate is the probability that the TD signal crosses the x-axis from one sample to another. It is calculated per time frame [JV04]:

$$x_{\mathrm{zcr}}(l) = \frac{1}{N-1} \sum_{n=lF+1}^{lF+N-1} \frac{1}{2} |\mathrm{sign}(s_{\mathrm{nb}}(n-1)) - \mathrm{sign}(s_{\mathrm{nb}}(n))|, \tag{6.11}$$

where $\text{sign}(x)$ is the sign operation

$$\text{sign}(x) = \begin{cases} 1 & \text{for } x > 0 \\ 0 & \text{for } x = 0 \\ -1 & \text{for } x < 0 \end{cases} . \tag{6.12}$$

The TD signal of voiced frames is often dominated by the slow oscillation of the pitch frequency which leads to a low zero crossing rate. Unvoiced frames without tonal elements have a noisy excitation with a higher probability of zero crossings. That is why the zero crossing rate has also been used as voicing criterion in other approaches [JV04].

Features that were proposed in the related paper [SFS18]:

- **Onset probability**: The beginning of a word after a short time of silence in a speech signal or a suddenly increasing energy in the speech signal is in the following named onset. Onsets can be detected in the magnitude spectrum by comparing the difference between two adjacent time frames with a threshold value:

$$d_{\text{ons}}(k,l) = \begin{cases} 1 & \text{for} \quad S^{\Delta}_{\text{nb}}(k,l) > \theta_{\text{ons}} \\ 0 & \text{otherwise} \end{cases} , \tag{6.13}$$

with the delta magnitude

$$S^{\Delta}_{\text{nb}}(k,l) = |S_{\text{nb}}(k,l)| - |S_{\text{nb}}(k,l-1)|. \tag{6.14}$$

Calculating the onset probability for each sub-band of the filterbank would be inefficient because the input feature vector of the network would get very long. However, depending on the speech signal and the background noise, onsets might be detected more reliably in some frequency ranges and it might be of interest whether the onset occurred in high or low frequencies of the NB spectrum. In informal training experiments, it was found to be a good trade-off to divide the NB spectrum into 4 frequency bands of 1 kHz bandwidth [SFS18]. Given a frequency range with the band indices $\{k_{\text{ons}} \cdot {}^{(K-1)}/_{4}, \ldots, (k_{\text{ons}}+1) \cdot {}^{(K-1)}/_{4}\}$, the onset probability for the band with index k_{ons} computes to

$$x_{\text{ons}}(k_{\text{ons}}, l) = \frac{4}{K} \sum_{k=k_{\text{ons}} \cdot {}^{(K-1)}/_{4}}^{(k_{\text{ons}}+1) \cdot {}^{(K-1)}/_{4}} d_{\text{ons}}(k,l), \tag{6.15}$$

with $k_{\text{ons}} \in \{0, \ldots, 3\}$.

- **Offset probability**: An offset will be detected if the magnitude of a given frequency bin falls for at least a defined threshold value from one frame to another. It can be calculated like eq. (6.15) to

$$x_{\text{off}}(k_{\text{off}}, l) = \frac{4}{K} \sum_{k=k_{\text{off}} \cdot {}^{(K-1)}/_{4}}^{(k_{\text{off}}+1) \cdot {}^{(K-1)}/_{4}} d_{\text{off}}(k,l), \tag{6.16}$$

where the delta magnitude $S_{\text{nb}}^{\Delta}(k,l)$ is multiplied with -1 in eq. (6.13):

$$d_{\text{off}}(k,l) = \begin{cases} 1 & \text{for} \quad -S_{\text{nb}}^{\Delta}(k,l) > \theta_{\text{off}} \\ 0 & \text{otherwise} \end{cases} . \tag{6.17}$$

Again, the number of coefficients can be chosen arbitrarily and is set to 4 in this work.

- **Signal above noise**: This feature was thought to give a robust hint whether the speech part dominates the signal [SFS18]. The dominance is evaluated for each frequency band and finally averaged in order to obtain a probability value. The spectral precision can be increased by averaging over parts of the spectrum separately. Like for the onset and offset calculation, the spectrum is split up in four equally sized parts $\{k_{\text{san}} \cdot {}^{(K-1)}/_{4}, \ldots, (k_{\text{san}}+1) \cdot {}^{(K-1)}/_{4}\}$ with the indices $k_{\text{san}} \in \{0, \ldots, 3\}$. For every sub-band, the feature indicates the probability that the SNR exceeds a threshold value θ_{san}:

$$x_{\text{san}}(k_{\text{san}}, l) = \frac{4}{K} \sum_{k=k_{\text{san}} \cdot {}^{(K-1)}/_{4}}^{(k_{\text{san}}+1) \cdot {}^{(K-1)}/_{4}} d_{\text{san}}(k,l), \tag{6.18}$$

with

$$d_{\text{san}}(k,l) = \begin{cases} 1 & \text{for SNR}(k,l) > \theta_{\text{san}} \\ 0 & \text{otherwise,} \end{cases} \tag{6.19}$$

where $\text{SNR}(k,l)$ denotes the signal-to-noise ratio in frame l and sub-band k.

6.2.3 Comparison of TD and FD Features

The TD features showed a high sensitivity to background noise in informal studies. The background noise can be reduced by a noise reduction algorithm before the feature extraction. Many recent noise reduction algorithms run in the FD, like the ABE algorithm in this work. Because of this, the TD features shall be replaced by FD features in this chapter so that the output spectrum from the noise reduction can be used directly as input for the feature extraction. Another advantage of FD-based features is that they can be more robust against strong interferences in small frequency ranges, e.g. by calculating the mean value over all sub-bands. In the following, the three TD features local kurtosis, gradient index, and zero crossing rate will be replaced by suitable FD features. All features are plotted for an exemplary sentence in fig. 6.1, once for the clean signal and once for a noisy version at an SNR of 10 dB.

Local kurtosis The local kurtosis is known as a measure for onsets and for the voiced-/unvoiced-decision [Jax02]. The onset feature can therefore not fully replace the local kurtosis. However, other FD features that were created for the voiced-/unvoiced-decision did not perform well in the feature selection approach. Like explained above,

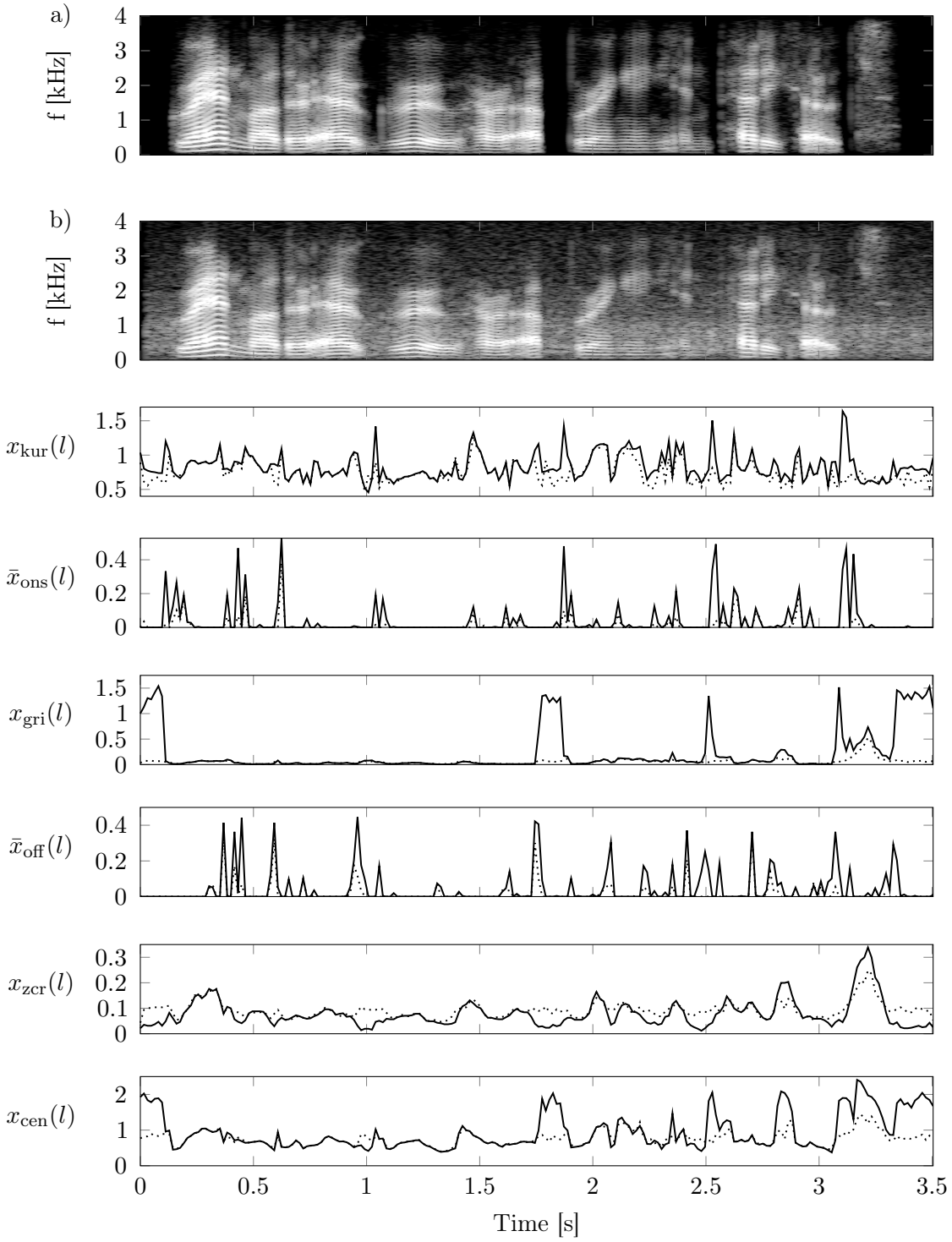


Figure 6.1: NB spectrogram of a speech signal from the TIMIT database with a comparison of several features. The clean NB spectrogram is depicted in (a). A noisy version, in which driving noise is added with an SNR of 10 dB, is shown in (b). The following graphs show the local kurtosis, the onset feature, the gradient index, the offset feature, the zero crossing rate and the centroid. The dotted graphs refer to the noisy signal while the solid graphs represent the clean speech.

the onset feature is calculated in 4 frequency ranges. In fig. 6.1, just the average of these 4 values

$$\bar{x}_{\text{ons}}(l) = \frac{1}{4} \sum_{k_{\text{ons}}=0}^{3} x_{\text{ons}}(k_{\text{ons}}, l) \tag{6.20}$$

is depicted in order to keep the complexity low. It can be seen by inspecting the dotted graphs that some peaks vanish after adding noise to the signal for both features, which means that they are both sensitive to additional background noise. With a noise reduction that runs before the feature extraction, the FD-based feature can be enhanced without having to perform ISTFT and STFT again.

Gradient index The gradient index is a scalar value that can be used as an indicator for unvoiced speech [Jax02]. It is detected in those frames where the gradient of the TD signal changes its sign frequently. As it can be observed in fig. 6.1, this works best for clean speech. The background noise makes the feature hard to interpret. Again, there is no FD feature that fully replaces the gradient index. A combination of the centroid and the offset feature might yield most of the relevant information. The offset feature detects the time frames, in which the speech level drops rapidly, e.g. at the beginning of a speech pause. This is why some peaks of the offset feature are aligned with rising slopes of the gradient index. The centroid is low for vowels where the main energy is located in the lower frequencies. Although it does not take the fine structure of the harmonics into account, its values are similar to those of the gradient index. The centroid is also sensitive to noise when the SNR is low, it then depends on the energy distribution of the background noise. A noise reduction can alleviate this effect.

Zero crossing rate The zero crossing rate can be used to detect fricatives as it detects frames where the signal changes its sign frequently. Especially frames where the main energy is located in higher frequencies have a lot of zero crossings. The centroid can again be used as a replacement that behaves similar in most cases. In the example in fig. 6.1, it can be observed that frames with silence can build an exception. The centroid has high values in these frames as the energy distribution is rather flat while the zero crossing rate is low. However, the detection of speech pauses can be done using a simple power-based feature like the signal power level described above.

6.2.4 Forward Selection for ABE

In a related work [SFS18], the size of the feature pool was chosen to be equal to 10 which allowed for using a simple wrapper approach. Forward selection and backward elimination are two well known wrapper methods [KJ97]. The forward selection method was chosen in [SFS18] as it is less expensive regarding computation time. The implementation for ABE in this work is explained briefly in the following.

After the initialization, the feature set is empty and the feature pool consists of all 10 features from table 6.1. In an iterative process, the feature set is built by adding one

feature from the pool in each iteration. The selection of the feature that shall be added is done based on the evaluation of a set of trained DNNs that only differ in their input features. In the first iteration, 10 DNNs are trained, each with one of the 10 input features from the pool. The feature that leads to the best DNN performance is selected and is moved from the feature pool to the input feature set. In all following iterations, the feature set from the last iteration is combined one by one with all features from the pool. Again, the feature set that leads to the best performance is selected. The process is stopped when the performance improvement drops below a threshold. The threshold controls the trade-off between good performance and low complexity.

Each time a feature is added to the feature vector, it is replaced by its respective Δ-feature in the pool. Similarly, Δ-features are replaced by $\Delta\Delta$-features. All Δ- and $\Delta\Delta$-features have the same dimensionality as the respective features of the current frame. The only exception is the MFCC feature with 30 coefficients for standard features, 20 for Δ-features and 10 for $\Delta\Delta$-features. The dimensionality is reduced by taking only the first 20/10 cepstral coefficients, which are already sufficient for a rough estimation of the spectral envelope. The reduction is motivated by the high computational cost of a large input feature vector and the lower relevance of the last coefficients of the cepstrum. The first 10 or 20 coefficients are taken as these are already sufficient for an approximation of the envelope.

The loss of a trained DNN for a given input feature after convergence is chosen as quality measure. A focus lies on the robustness of FD features in comparison with TD features regarding background noise and channel characteristics. Accordingly, various simulated transmission channels and noise scenarios were simulated in the data generation process for the training and validation dataset.

A drawback of this implementation of a wrapper method is that the loss function does not always correlate well with the subjective quality. Because of this, a second feature selection approach was applied to another target. As stated above already, a main difficulty in ABE is to distinguish between sibilant fricatives ([s], [z], [ʃ], and [ʒ]) and other phonemes (see section 3.1 and [BAF14]). If the inserted UB energy of the sibilant fricatives [s] and [z] is not high enough, it seems that the speaker has a lisp. To evaluate the ability of a DNN to detect sibilant fricatives based on different input feature sets, the feature selection experiment was additionally conducted on a classification task. Classification DNNs were trained to distinguish between the two classes *sibilant fricative* and *other phoneme*. Only those features that perform well in both tasks can reliably yield a good perceived quality after the ABE training.

6.3 DNN Training Setup

The basic network which is used in this thesis is a regression MLP named R, which is a feedforward DNN architecture (see section 2.1). The target feature vector consists of 30 MFCCs that represent the WB spectral envelope for each frame, based on 40 mel bands. The standard loss function $J(l, \Theta)$ is the MSE between the true and the

predicted output feature vectors, $\mathbf{y}(l)$ and $\hat{\mathbf{y}}(l, \Theta)$. Each time frame, a feature vector is extracted from the short-term spectrum of the NB signal. L feature vectors are concatenated to one mini-batch with index m for the calculation of the loss function and form the feature matrices $\mathbf{Y}(m) = [\mathbf{y}(mL), \ldots, \mathbf{y}((m+1)L-1)]$ and $\hat{\mathbf{Y}}(m, \Theta) = [\hat{\mathbf{y}}(mL, \Theta), \ldots, \hat{\mathbf{y}}((m+1)L-1, \Theta)]$.

The updates of the network parameters are performed based on the widely known backpropagation algorithm explained in section 2.2.2. In an iterative process, the DNN's parameters Θ are updated to minimize the loss function after each mini-batch of data. The training and the validation loss are evaluated for each epoch of the training process. The DNN that achieves the lowest validation loss is used later in the prediction stage. If the validation loss does not decrease for 30 successive epochs, it is assumed that it might not decrease further and the training process is stopped.

6.4 Discriminative Training

Many approaches for DNN-based ABE use regression networks for the extension of the spectral envelope. As a loss function during training, mostly the MSE between true and estimated wideband spectral envelopes is used. One of the problems with MSE training that is commonly known is the problem of *over-smoothing*. This means that the network learns to predict a smoothed version of the targets with less variance over time. As stated in section 3.1, sibilant fricatives have much more energy in the UB than all other phonemes. However, a DNN that is trained on just the MSE loss function tends to extend all phonemes similarly and to underestimate the differences between different phoneme classes. As a consequence, the extension energy is generally overestimated for vowels and underestimated for sibilant fricatives like [s] and [z]. The overestimation leads to disturbing artifacts for vowels and the underestimation to a lisping effect that reduces the speech quality significantly [BF09].

In order to reduce this effect, a modification of the loss function is proposed in [Sau+18b]. It is explicitly taken into account in the training that sibilant fricatives have to be extended more strongly than other phonemes in order to achieve ABE results with a high perceived quality. For this, a mean energy ratio between the UB energy of sibilant fricatives and the UB energy of all other phonemes is calculated on the true WB training data. Deviations of this ratio introduced by the prediction are punished in the loss function while training the DNN. As the perceived speech quality highly depends on the correct extension of sibilant fricatives [BF09], this improves the speech quality of the extended signal. The evaluation of the quality improvement by discriminative training in [Sau+18b] was done through subjective listening tests (see section 7.5).

In this section, a discriminative term $\bar{J}^{\text{dis}}(m, \Theta)$ is added to the basic MSE loss function $\bar{J}^{\text{mse}}(m, \Theta)$. This term shall force the network to preserve the power ratio

between sibilant fricatives and other phonemes. A general formulation of this is

$$\bar{J}^{\text{mse,dis}}(m,\Theta) = \gamma^{\text{mse}}\bar{J}^{\text{mse}}(m,\Theta) + \gamma^{\text{dis}}\bar{J}^{\text{dis}}(m,\Theta), \tag{6.21}$$

where γ^{mse} and γ^{dis} are weights that trade-off the MSE loss function versus the discriminative term. According to section 6.3, the output feature vector $\mathbf{y}(l)$ consists of 30 MFCCs. The predicted MFCCs are transformed back to the logarithmic mel-based PS

$$\hat{\Phi}^{[\text{dB}]}_{S'S',\text{wb}}(k',l,\Theta) = \text{IDCT}(\hat{c}_{\text{wb}}(i_c,l,\Theta)) \tag{6.22}$$

by an IDCT like in eq. (3.10). Transforming these logarithmic values back to the power domain allows for calculating the sum over frequency bands:

$$\hat{\Phi}_{S'S',\text{wb}}(k',l,\Theta) = 10^{\hat{\Phi}^{[\text{dB}]}_{S'S',\text{wb}}(k',l,\Theta)/10}. \tag{6.23}$$

As only the UB has to be estimated in the ABE algorithm, the UB mel-based PS is obtained by setting the lower 30 of all 40 mel-bands to zero:

$$\Phi_{S'S',\text{ub}}(k',l) = \begin{cases} \Phi_{S'S',\text{wb}}(k',l) & \text{for } k' \geq 30 \\ 0 & \text{otherwise} \end{cases}. \tag{6.24}$$

The calculation of the mel-based mean UB power level[1] can then be done similarly to the WB case but over the last 10 mel bands:

$$p'_{\text{wb}}(l) = \frac{1}{K'}\sum_{k'=0}^{K'-1}\Phi_{S'S',\text{wb}}(k',l) \tag{6.25}$$

$$p'_{\text{ub}}(l) = \frac{1}{10}\sum_{k'=30}^{K'-1}\Phi_{S'S',\text{ub}}(k',l). \tag{6.26}$$

The prediction of the UB power level $\hat{p}'_{\text{ub}}(l,\Theta)$ is calculated accordingly, based on the predicted mel-based PS $\hat{\Phi}_{S'S',\text{ub}}(k',l,\Theta)$.

As the differentiation between sibilant fricatives and other phonemes is a major problem in ABE without discriminative training, two phoneme classes were defined: one class with {[s], [z], [ʃ], [ʒ]} and one with all other phonemes (for more details, see section 3.1). In order to be able to use this information in the loss function, the detection of a sibilant fricative is defined as follows:

$$d_{\text{sfr}}(l) = \begin{cases} 1 & \text{if the phoneme at frame } l \text{ is in } \{[\text{s}], [\text{z}], [\text{ʃ}], [\text{ʒ}]\} \\ 0 & \text{otherwise.} \end{cases} \tag{6.27}$$

The UB power ratio between the two phoneme classes is called sibilant fricative power ratio (SFPR) [Sau+18b] and computes to:

$$q(m) = \frac{\sum_{l=mL}^{(m+1)L-1} p'_{\text{ub}}(l)\cdot d_{\text{sfr}}(l)}{\sum_{l=mL}^{(m+1)L-1} p'_{\text{ub}}(l)\cdot(1-d_{\text{sfr}}(l))} \tag{6.28}$$

[1]The mel-based mean UB power level is calculated on the mel magnitude which inserts a deviation compared to the UB power level calculated on the spectral magnitude. The difference in the notation is the apostrophe that indicates mel-based measures.

The SFPR of the prediction $\hat{q}(m, \Theta)$ is calculated similarly by replacing the true UB power level $p'_{\mathrm{ub}}(l)$ by the predicted UB power level $\hat{p}'_{\mathrm{ub}}(l, \Theta)$. The implementation of the SFPR can have its difficulties when dealing with a small batch size. When the batch size is low, it is not guaranteed that both classes occur in every batch which means that a division by zero could occur. A way to avoid this is to choose a relatively high batch size and set the SFPR to zero when one of the classes is not present.

The deviation between the SFPR that is calculated on the predicted envelopes $\hat{q}(m, \Theta)$ and the SFPR calculated on the real WB data $q(m)$ can be formulated as a loss function which shall be minimized in the training process. This is done by the discriminative term of the loss function, which was defined as the squared relative error of the SFPR [Sau+18b]:

$$\bar{J}^{\mathrm{dis}}(m, \Theta) = \left| \frac{q(m) - \hat{q}(m, \Theta)}{q(m)} \right|^2 . \tag{6.29}$$

The relative error has the advantage that the SFPR does not have to be normed regarding the probabilities of both phoneme classes as they appear equally often in the nominator and the denominator. With suitable weights γ^{mse} and γ^{dis} from eq. (6.21), the DNN is forced to approximately reproduce the original SFPR in addition to reducing the MSE. Consequently, the UB energy of the predicted envelope would be high enough for sibilant fricatives and low enough for other phonemes like vowels, each compared to the respective WB training data.

6.5 Adversarial Training

The motivation for discriminative training was the over-smoothing effect of MSE-based DNN training, which manifests in strongly underestimated dynamics of the UB [Sau+18b]. The results in chapter 7 show that discriminative training reduces this effect but they also show that the problem is not completely solved. As the over-smoothing effect originates from the MSE training, other training methods could perform better on a regression task like UB envelope estimation. One training method, which does not use the MSE as cost function, is the training of generative adversarial networks (GANs) (see section 2.5). Adversarial training is in this section applied to the training of WB MFCCs for ABE.

GANs have been applied first to image processing tasks [Iso+16; Led+16] where they proved their high potential. More and more, they have also been applied to other fields of research. In the field of speech processing, GANs have already successfully been applied to speech enhancement tasks [DLP17; PBS17; Wan+18]. In statistical parametric speech synthesis, they could solve the problem of over-smoothing [Kan+17; STS18]. This suggests that they might also solve the over-smoothing problem that occurs in ABE when training a feedforward DNN. In 2018, Li et al. showed that GANs outperform codebook and HMM approaches on the ABE task [Li+18]. However, a similar comparison has already been evaluated with another DNN architecture in [AF17]

that also outperformed the classical HMM-based approaches. The comparison between a basic feedforward DNN and a comparable GAN gives a better insight and is covered in this thesis.

The original GAN structure mainly has the target of generating data from random noise. In this thesis, the problem is a regression task where input features are given and the outputs must fit to these inputs. In order to better use this knowledge in the training structure, a conditional GAN (CGAN) is applied to ABE in section 6.5.2.

Discriminative training and CGAN training are two different approaches with the same aim, the reduction of over-smoothing. Given that these methods act in a different way, a combination of both methods together with MSE training might lead to a further reduction. Section 6.5.3 shows the approach of combining adversarial training and discriminative training based on earlier studies [Sau+19]. The results are evaluated in section 7.6.

6.5.1 GAN Training

The basics of GAN training are explained in section 2.5, this section is about the application of GANs to ABE. The network architecture for GAN training contains two networks, namely a generator and a discriminator network. The generator network G is given by the regression network R from section 6.3 and the discriminator network D is created (see fig. 2.6a). R and G are identical in their structure, the different naming just makes clear that the network G is trained in a GAN together with a discriminator D. Another difference to the MSE training is that, in contrast to the training of R, no pre-training was applied for G. The structure is chosen to be identical in order to yield comparable results in the different approaches.

In section 2.5, it is explained that the generator receives random noise $\mathbf{z}(l)$ as input in the basic GAN approach [Goo+14]. For ABE, a NB feature vector $\mathbf{x}(l)$ replaces the random noise [Li+18]. The noise input is not necessary because a single, optimal solution is defined for a given NB feature vector. The output of G is the WB feature vector

$$\hat{\mathbf{y}}(l, \Theta^G) = G(\mathbf{x}(l), \Theta^G), \tag{6.30}$$

which corresponds to the predicted WB MFCCs as in the simple regression DNN R.

In some recent approaches [Li+18; Iso+16; Pat+16], the GAN loss function from eq. (2.31) has been successfully combined with the standard MSE loss function from eq. (2.11). This approach stabilizes the training process as the updates for the generator do not only depend on the discriminator any more. With the MSE loss, it is ensured that the final solution is to some extent similar to the real envelope. The details which are hard to learn with just the MSE loss function can then be learned by the adversarial loss. Following these approaches, the combined objective for GAN training [Iso+16]

$$\min_G \max_D \gamma^{\text{mse}} \bar{J}^{\text{mse}}(m, \Theta^G) + \gamma^{\text{gan}} \bar{J}^{\text{gan}}(m, \Theta^G, \Theta^D) \tag{6.31}$$

is used as definition. The optimal convergence state is reached when the prediction of G yields such a high quality that D cannot distinguish any more between true and generated samples.

6.5.2 CGAN Training

Originally, the GAN was designed to generate data from random noise. In ABE, however, the estimation problem is an input-dependent regression task, in which NB features shall be mapped to WB features. Therefore, the GAN from [Li+18] was replaced by a CGAN in [Sau+19]. CGANs are designed to train the discriminator with the inputs and the targets of the generator as inputs (see section 2.5), which are here the NB and the WB features, respectively. This lets the discriminator learn the conditional classification task whether its input WB data is real or generated, dependent on the given NB data. More specifically, $D(\mathbf{y}(l))$ and $D(G(\mathbf{x}(l)))$ are replaced by $D_\mathrm{c}(\mathbf{x}(l), \mathbf{y}(l))$ and $D_\mathrm{c}(\mathbf{x}(l), G_\mathrm{c}(\mathbf{x}(l)))$, as shown in fig. 2.6b. Apart from this difference, the training objective is identical:

$$\min_{G_\mathrm{c}} \max_{D_\mathrm{c}} \gamma^{\mathrm{mse}} \bar{J}^{\mathrm{mse}}(m, \Theta^{G_\mathrm{c}}) + \gamma^{\mathrm{cgan}} \bar{J}^{\mathrm{cgan}}(m, \Theta^{G_\mathrm{c}}, \Theta^{D_\mathrm{c}}). \tag{6.32}$$

6.5.3 Discriminative CGAN Training

The combination of discriminative training and CGAN training is achieved by simply adding the weighted loss terms. Specifically, the discriminative loss term $\bar{J}^{\mathrm{dis}}(m)$ from eq. (6.29) is added to the combined objective of GAN training in eq. (6.32) with the CGAN loss function from eq. (2.32):

$$\min_{G_\mathrm{c}} \max_{D_\mathrm{c}} \gamma^{\mathrm{mse}} \bar{J}^{\mathrm{mse}}(m, \Theta^{G_\mathrm{c}}) + \gamma^{\mathrm{cgan}} \bar{J}^{\mathrm{cgan}}(m, \Theta^{G_\mathrm{c}}, \Theta^{D_\mathrm{c}}) + \gamma^{\mathrm{dis}} \bar{J}^{\mathrm{dis}}(m, \Theta^{G_\mathrm{c}}), \tag{6.33}$$

where the weights were set to $\gamma^{\mathrm{mse}} = 1$, $\gamma^{\mathrm{cgan}} = 0.1$, and $\gamma^{\mathrm{dis}} = 2$. The discriminator was trained with twice the learning rate of the respective generator training for all trainings of GANs and CGANs. G_c was initially trained independently of D_c for 2000 steps. This was done to achieve a good initial direction for the update of the generator network as the discriminator is in a random state at the beginning of the training. In informal tests, this method yielded a slightly faster and more robust convergence. After these 2000 steps, D_c and G_c were trained alternately. This means that always either the weights and biases of D_c were fixed while G_c was trained or the other way around.

Chapter 7

Evaluation

The quality or the performance of algorithms are assessed in this chapter by assigning numeric quality values to algorithms. These values can be obtained by calculating objective measures directly on the data or by a subjective evaluation that is performed by humans. In most applications of speech signal enhancement, the enhanced speech is finally presented to human listeners. This implies that the goal is to maximize the perceived subjective signal quality. The best way to evaluate the perceived quality of a speech signal is in many cases to run subjective listening tests. However, as such listening tests are expensive and time-consuming, it is beneficial to find objective measures that correlate well with the subjective ratings. This task has to be solved for every application separately. Especially the well known measures WB-PESQ [ITU05] and POLQA [ITU11b] are not reliably predicting the overall speech quality of an ABE-processed signal [Abe+17]. They were developed for speech enhancement tasks like noise reduction, and the correlation with the perceived quality of an ABE system is rather low [Abe+17]. As a consequence, objective instrumental measures cannot fully replace subjective listening tests for the evaluation of ABE algorithms [Bau+14; Pul+15; Möl+13]. The objective measures in this chapter are mostly thought to additionally show an objective effect that was generated by a specific modification of an algorithm.

The setup that was used to generate the results in this chapter is described in section 7.1. This covers the ABE setup, the data generation, and the training of the DNNs. Section 7.2 introduces some common metrics that can be used for the evaluation of tasks that are related to ABE. However, there is no objective measure for the overall speech quality. In the last sections, the proposed methods from chapters 5 and 6 are evaluated and the results are compared.

7.1 Evaluation Setup

This section describes the setup that was used for the evaluations related to this work. This includes the ABE approach and the generation of the training data, as well as the training of the DNN. The setup for the training process is slightly changing from one evaluation to another. In this section, especially those parameters that all setups

have in common are explained. The detailed configurations are given in the respective evaluations in sections 7.3 to 7.7.

7.1.1 Artificial Bandwidth Extension Setup

The ABE system is based on the block diagram in fig. 4.1. All of the approaches that are evaluated in this chapter use a common set of input and output features. The input feature vector $\mathbf{x}(l)$ of dimension 135 consists of 30 NB MFCCs, their first and second derivatives in time (Δ-features) as well as several other features. The input feature set is described in more detail in [SFS18]. All networks predict the spectral envelope represented as a vector of 30 WB MFCCs, like described in section 6.3. The DNNs are trained offline in a separate training stage, on a training data set.

All speech signals that were used in this thesis were first resampled to a sample rate of $f_s = 16\,\text{kHz}$. The processing was performed in frames of $N = 512$ samples which are shifted by $F = 256$ samples, yielding a relative overlap of two consecutive windows of 50%. The filterbank that was used for the transformation to the FD runs with a FFT length of 512 which fits to the frame length. In order to achieve perfect reconstruction, square-root Hann windows are chosen for STFT and ISTFT like described in section 3.4.1.

7.1.2 Training Data Generation

The training data should cover as many conditions of input data that can occur in the prediction stage as possible in order to yield a good prediction performance after DNN training. The approach of training with data in different conditions is explained in section 2.2.7 under the name multi-condition training. Two main tasks in the generation process of multi-condition training data are the choice of speech and noise data on the one hand and signal processing methods to modify this data artificially on the other hand. The whole process of training data generation like it was done for the following trainings is shown in fig. 7.1. The basic speech data and all the processing steps are defined in more detail in the following list:

- **Speech Data**
  The speech data was taken from the TIMIT corpus [Gar+93] of American English speech. It contains about 5.4 hours of speech data. 630 speakers were recorded while reading 10 short sentences each. Each speaker was assigned to one of the datasets for training, validation, and testing, according to the distribution of 60% training and 20% validation and testing. Around 10%–30% of the speech signals were selected twice so that a total of 7000–8000 sentences were used in the DNN trainings (7000 in sections 7.3 and 7.5 and 8000 in section 7.6). This corresponds to slightly more than 4 hours of data in the training dataset. If the same signal was chosen twice, it was processed differently so that the DNN would not receive exactly the same data twice in one epoch. The TIMIT dataset also includes

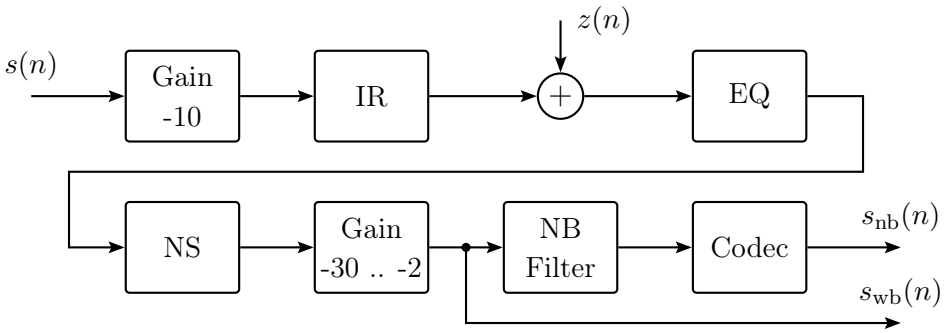


Figure 7.1: Block diagram of the training data generation process for a DNN that predicts the WB spectral envelope from features that are based on the NB signal. $z(n)$ is a background noise signal that is added to a processed version of the speech signal $s(n)$. IR denotes a convolution with an impulse response, EQ denotes a convolution with an equalizer filter, NS represents a noise suppression algorithm, and NB denotes a convolution with a narrowband filter. The gains are given in dBFS.

a phoneme transcription which was necessary for the phoneme-dependent loss functions.

- **Gain**
  The signals were amplified to a maximum amplitude of -10 dBFS (decibel full scale, level in dB compared to amplitude of 1) at first in order to prevent from clipping and from a low resolution of the amplitude in the next processing steps. Later in the data generation process, the signals were amplified to a random level between -30 and -5 dBFS (-30 and -2 dBFS in section 7.6). The levels in dBFS were chosen from a uniform random distribution in the logarithmic domain.

- **Impulse Response**
  50% of the signals were convolved with an impulse response (IR) that was randomly chosen from a predefined pool of impulse responses. The impulse responses originated from a collection of many different in-car room impulse responses (RIRs) and some other RIRs with a rather short reverberation time. Each impulse response models a transfer function between the speaker at the far end of the phone call and the far-end device. It contains the information about the room, the microphone characteristics, and the relative location of the speaker to the microphone. Especially the microphone characteristics can have a high influence on MFCCs because of a strong HP that is applied to some microphones.

- **Background Noise**
  For 50% of the speech signals, background noise was added to the signal (75% in section 7.6). This simulates the background noise situation at the far end. The overall SNR was chosen randomly from a logarithmic uniform distribution between 10 and 25 dB. Mostly stationary background noise scenarios that were recorded inside a car while driving were chosen.

- **Equalizer**
  Further variation introduced by varying characteristics of microphones and small changes of the room impulse response at the far end can be simulated by an equalizer (EQ) that randomly amplifies or attenuates some frequency ranges. In order to train the DNN to be robust against these changes, an additional EQ was applied to 25% of the processed signals. It was chosen randomly from a pool of several EQs that had been generated beforehand as superposition of gaussian distributions in the FD. Like this, the network might learn that a special EQ should not have an influence on the results of the prediction.

- **Noise Suppression**
  In the final implementation, the ABE algorithm might receive speech data that was already processed by a noise suppression algorithm. This makes it obvious that the DNN should also be trained with such data. A noise suppression that focuses on stationary noise was applied to all of the signals with a maximum attenuation of the noise floor that was randomly chosen from a logarithmic uniform distribution between 0 and 20 dB. This means that each time-frequency bin in the noisy spectrogram can at most be attenuated to the estimated stationary background noise minus the chosen value in dB.

- **Narrowband Filter**
  When the processing for the WB training data was completed, the WB signals were convolved with a NB filter. This filter was randomly chosen from a set of predefined BP filters with slightly varying cutoff frequencies around 300 Hz and 3.4 kHz and varying slopes. This models the behavior of different cellphones regarding the BP for NB signals in the sending direction.

- **Codecs**
  The codec with which the speech signal was transmitted in the cellular network also influences the speech signal. Also the rate at which the codec operates influences the quality strongly and can introduce a characteristic degradation. 75% of the NB speech signals were transcoded with a NB codec. The type of codec and the rate at which the codec operates are randomly chosen from the lists that are presented in section 3.4.4. These are some of the codecs that are widely used and may occur in real phone calls as well.

Although this list does not cover all variations that occur in real recordings, it helps the DNN to generalize well. Instead of putting a high effort into finding further ways of data modification, the generalization capabilities of DNNs can also be improved by regularization methods.

7.1.3 Training Process

Some of the parameters that have to be set for a DNN training are explained in the following. For the baseline regression DNN, the MSE loss function $J^{\text{mse}}(m, \Theta)$ is

used, while for GAN training, the combined objective from eq. (6.31) is used. In all experiments, the weight of the MSE loss function is set to $\gamma^{\text{mse}} = 1$. The weight for the discriminative trainings was set to $\gamma^{\text{dis}} = 2$ in section 7.5. A weight of $\gamma^{\text{gan}} = 0.1$ gave good results in preliminary experiments and has therefore been set in the adversarial trainings. The CGAN is trained like the basic GAN, just using the CGAN loss function from eq. (2.32) for the discriminator network instead of eq. (2.31). Accordingly, the weight is also set to $\gamma^{\text{cgan}} = 0.1$.

L2-regularization was performed with a weight between 0.002 and 0.004 in order to prevent from overfitting. Only in classification trainings, dropout training was applied. The experience showed that dropouts had mostly a negative impact on the results when applied to regression tasks. The Adam optimizer [KB14] was chosen for the DNN training which is widely used in recent ML approaches.

Architecture The network dimensions for the regression network R, the generator networks G and G_{c} and the discriminator networks D and D_{c} are the same. All have two hidden layers with 128 nodes per layer. Each layer is fully connected like in an MLP. All layers except the output layer are combined with a ReLU activation function. The input layer depends on the length of the input feature vector and is set to $N_{\mathbf{x}} = 135$. The output layer depends on the length of the target feature vector and is $N_{\mathbf{y}} = 30$ for regression networks and $N_{\mathbf{y}} = 2$ for classification networks in the scope of this thesis. The initialization of the weights was done following a random distribution for most trainings, only the weights in section 7.6 were pre-trained using a stacked auto-encoder (see section 2.3).

The only trainings that differ in their architecture are the trainings that were done for the input feature selection in section 7.3. They had just 64 nodes per layer and the input feature dimension rises in every iteration of the algorithm.

Hyperparameters The mini-batch size was set to 128 for the trainings in sections 7.3 and 7.5 and to 192 for the trainings in section 7.6. The initial learning rate was set to $\eta = 3 \cdot 10^{-6}$ in section 7.5 and to $\eta = 1 \cdot 10^{-5}$ in sections 7.3 and 7.6. For the L2 regularization weight, $\gamma^{\text{reg}} = 4{\cdot}10^{-3}$ was applied in sections 7.5 and 7.6 and $\gamma^{\text{reg}} = 2{\cdot}10^{-3}$ in section 7.3. An auto-encoder was only used in the regression DNN in section 7.6, as it did not improve the results for GAN trainings.

7.2 Evaluation Metrics

The aim of the evaluation is to compare the quality of the ABE algorithms. Additionally, it is used to check whether some related tasks can be solved by the algorithms. This section presents the metrics that are used in the subjective and objective evaluations in the following sections. As human perception is the most important quality measure in ABE, a rating scale has to be defined. Normed rating scales for subjective evaluation are presented in section 7.2.1. Because of the high costs of subjective listening tests,

5	excellent
4	good
3	fair
2	poor
1	bad

(a) MOS

5	inaudible
4	audible but not annoying
3	slightly annoying
2	annoying
1	very annoying

(b) DMOS: 'degradation is ...'

3	much better
2	better
1	slightly better
0	about the same
-1	slightly worse
-2	worse
-3	much worse

(c) CMOS

Table 7.1: Rating scales for subjective listening tests according to ITU-T recommendation P.800 [ITU96a].

objective measures were developed that predict the subjective scores. Those measures that are used in this thesis are defined in section 7.2.2.

7.2.1 Subjective Metrics

For subjective listening tests, a normed rating scheme has to be defined in order to be able to compare results between different algorithms. Rating scores for speech quality assessment have been defined by the ITU in [ITU96a]. Three main rating methods were defined:

- **ACR**: In the absolute category rating (ACR) method, each speech sample is rated separately with values from 1 to 5 that are mapped to a subjective quality expression in table 7.1a. The unit of the numerical rating is mean opinion score (MOS).
- **DCR**: In the degradation category rating (DCR) method, a speech sample A is compared to a speech sample B. B is in this case a processed version of A. Every processing can introduce a degradation in terms of speech quality. This degradation is measured using 1 to 5 degradation mean opinion score (DMOS) points from table 7.1b where 5 means that no changes are audible.
- **CCR**: In the comparison category rating (CCR) method, two speech samples A and B are compared. In contrast to the DCR method, A and B can be arbitrary signals and the order of A and B is randomly varying. The rating scale ranges from -3 (A much worse than B) up to 3 (A much better than B) in comparison mean opinion score (CMOS) points. The scores with their respective meanings are shown in table 7.1c.

The results of the listening tests are mainly evaluated based on the mean ratings

$$\bar{r} = \frac{1}{N_{\mathbf{r}}} \sum_{i_r=0}^{N_{\mathbf{r}}-1} r(i_r), \tag{7.1}$$

where $r(i_r)$ is the rating with the rating index i_r, and $N_\mathbf{r}$ denotes the number of ratings. But there are more metrics that can be extracted from the rating results, which will be introduced in the following. The standard deviation of the mean, also called the standard error, can be estimated to

$$\sigma_{\bar{r}} = \sqrt{\frac{\sigma_r^2}{N_\mathbf{r}}}, \tag{7.2}$$

where σ_r^2 denotes the variance of the rating vector $\mathbf{r}$. As the number of ratings is rather small, the unbiased sample variance should be determined using the Bessel's correction:

$$\sigma_r^2 = \frac{1}{N_\mathbf{r} - 1} \sum_{i_r=0}^{N_\mathbf{r}-1} (r(i_r) - \bar{r})^2. \tag{7.3}$$

Most of the listening tests that were conducted as basis for this thesis are CMOS tests with comparisons of two signals A and B. To check the reliability of each listener in a CMOS test, a measure called mean difference voting ratio (MDVR) was proposed in a prior work [Sau+18a]. It divides the mean absolute voting of all A-A comparisons $\bar{r_{\mathrm{AA}}}$ through the mean absolute voting of all A-B comparisons $\bar{r_{\mathrm{AB}}}$ of a listener:

$$MDVR = \frac{\bar{r_{\mathrm{AA}}}}{\bar{r_{\mathrm{AB}}}} = \frac{\frac{1}{N_{\mathbf{r}_{\mathrm{AA}}}} \sum_{i_r=0}^{N_{\mathbf{r}_{\mathrm{AA}}}-1} |r_{\mathrm{AA}}(i_r)|}{\frac{1}{N_{\mathbf{r}_{\mathrm{AB}}}} \sum_{i_r=0}^{N_{\mathbf{r}_{\mathrm{AB}}}-1} |r_{\mathrm{AB}}(i_r)|}. \tag{7.4}$$

Here, $N_{\mathbf{r}_{\mathrm{AA}}}$ denotes the length of the vector $\mathbf{r}_{\mathrm{AA}} = [r_{\mathrm{AA}}(0), \ldots, r_{\mathrm{AA}}(N_{\mathbf{r}_{\mathrm{AA}}})]^\mathsf{T}$ and $N_{\mathbf{r}_{\mathrm{AB}}}$ denotes the length of the vector $\mathbf{r}_{\mathrm{AB}}$ which is defined accordingly. An MDVR larger than 1 means that a listener heard more differences between two equal sound signals than between different sound signals. This can be taken as a hint that the rating was not done reliably or that the differences between different signals were nearly inaudible.

7.2.2 Objective Metrics

Every comparison between ABE algorithms was additionally evaluated using objective measures that shall complement the results of the subjective listening tests. These objective measures are listed in the following paragraphs.

MSE The mean square error (MSE) is used as loss function in many regression tasks like estimating the WB spectral envelope for ABE. Although the MSE is widely used, there are also drawbacks like the over-smoothing effect. The evaluation of the MSE on different datasets can reveal whether a DNN training generalizes well by comparing the errors. The calculation of the MSE is given in eq. (2.11).

MSE on Logarithmic Mel-Based UB PSD In the regression DNNs used in this thesis, the target feature always consists of WB MFCCs. Converting the coefficients

back to mel-band spectra makes the deviations easier to interpret. The MSE between the true and the predicted logarithmic mel-based PSs in the UB

$$e_{\text{ub}}^{\text{mse}\prime}(l,\Theta) = \sum_{k'=0}^{K'-1} \left| \Phi_{S'S',\text{ub}}^{[\text{dB}]}(k',l) - \hat{\Phi}_{S'S',\text{ub}}^{[\text{dB}]}(k',l,\Theta) \right|^2 \tag{7.5}$$

is used in this chapter in the objective evaluation of the convergence of regression tasks. The predicted mel-based UB PS $\hat{\Phi}_{S'S',\text{ub}}^{[\text{dB}]}(k',l,\Theta)$ is calculated based on the predicted WB MFCC feature vectors $\hat{c}_{\text{wb}}(i_c,l,\Theta)$ according to eq. (3.10) and eq. (6.24).

Mel-Based LSD The deviation between two spectrograms can be measured as log-spectral distance (LSD). While the standard LSD is calculated on the whole WB spectrum, the same measure can also be formulated based on the UB part of the spectrum. This allows to focus on those frequencies that are estimated by the ABE algorithm. The LSD of frame l based on the predicted mel-based PS can be written as:

$$e_{\text{wb}}^{\text{lsd}\prime}(l,\Theta) = \sqrt{\sum_{k'=0}^{K'-1} \left| \Phi_{S'S',\text{wb}}^{[\text{dB}]}(k',l) - \hat{\Phi}_{S'S',\text{wb}}^{[\text{dB}]}(k',l,\Theta) \right|^2}. \tag{7.6}$$

The LSD of the UB, $e_{\text{ub}}^{\text{lsd}\prime}(l,\Theta)$, is calculated like $e_{\text{wb}}^{\text{lsd}\prime}(l,\Theta)$, just based on the logarithmic mel-based PSs of the UB:

$$e_{\text{ub}}^{\text{lsd}\prime}(l,\Theta) = \sqrt{\sum_{k'=0}^{K'-1} \left| \Phi_{S'S',\text{ub}}^{[\text{dB}]}(k',l) - \hat{\Phi}_{S'S',\text{ub}}^{[\text{dB}]}(k',l,\Theta) \right|^2}. \tag{7.7}$$

Relative Error of the Mean UB Power With the mean UB power, it can be evaluated whether the prediction is generally too strong or too weak. The predicted mean UB power

$$\hat{\mu}_{\text{ub}}(m,\Theta) = \frac{1}{L} \sum_{l=mL}^{(m+1)L-1} \hat{p}'_{\text{ub}}(l), \tag{7.8}$$

with $\hat{p}'_{\text{ub}}(l)$ as defined in eq. (6.26), is compared to the true mean UB power in the formula for the relative error:

$$e_{\text{ub}}^{\mu}(m,\Theta) = \frac{\hat{\mu}_{\text{ub}}(m,\Theta) - \mu_{\text{ub}}(m)}{\mu_{\text{ub}}(m)}. \tag{7.9}$$

The alternative calculation in dB is based on the mean UB power values in dB and does not norm the error to the real mean value:

$$e_{\text{ub}}^{\mu[\text{dB}]}(m,\Theta) = \hat{\mu}_{\text{ub}}^{[\text{dB}]}(m,\Theta) - \mu_{\text{ub}}^{[\text{dB}]}(m). \tag{7.10}$$

Relative Error of the Standard Deviation of the UB Power There is no general rule for the amount of energy that has to be inserted in ABE in the UB. But at least it can be observed that some phoneme classes like sibilant fricatives need to be extended more strongly than others. By assigning a mean UB energy value to each phoneme class, the differences between phoneme classes can be evaluated. These differences can have a high impact on the perceived speech quality. When over-smoothing occurs, the differences are too small. This effect can be quantified in the standard deviation $\hat{\sigma}_{\mathrm{ub}}$ of the predicted power level $\hat{p}'_{\mathrm{ub}}(l)$ (see eq. (6.26)) over time

$$\hat{\sigma}_{\mathrm{ub}}(m,\Theta) = \frac{1}{L-1} \sum_{l=mL}^{(m+1)L-1} (\hat{p}'_{\mathrm{ub}}(l) - \hat{\mu}_{\mathrm{ub}}(m,\Theta))^2. \tag{7.11}$$

The relative error of this standard deviation of the predicted UB power level

$$e^{\sigma}_{\mathrm{ub}}(m,\Theta) = \frac{\hat{\sigma}_{\mathrm{ub}}(m,\Theta) - \sigma_{\mathrm{ub}}(m)}{\sigma_{\mathrm{ub}}(m)} \tag{7.12}$$

shows whether the prediction-based standard deviation is too low ($e^{\sigma}_{\mathrm{ub}}(m,\Theta) < 0$) or too high ($e^{\sigma}_{\mathrm{ub}}(m,\Theta) > 0$). An optimal value of 0 means that the variance of the energy distribution over time is modeled correctly. The alternative calculation in dB is based on the standard deviations in dB and does not norm the error to the real standard deviation:

$$e^{\sigma[\mathrm{dB}]}_{\mathrm{ub}}(m,\Theta) = \hat{\sigma}^{[\mathrm{dB}]}_{\mathrm{ub}}(m,\Theta) - \sigma^{[\mathrm{dB}]}_{\mathrm{ub}}(m) \tag{7.13}$$

Relative Error of the SFPR The discriminative term that was defined in eq. (6.29) depends on deviations of the SFPR (see eq. (6.28)) between estimated and true WB speech. These deviations can be evaluated as relative error of the SFPR:

$$e^{q}(m,\Theta) = \frac{\hat{q}(m,\Theta) - q(m)}{q(m)}. \tag{7.14}$$

Error Rate of Classification Task The error rate of a classification task is the averaged value of the frame-wise decision whether a prediction was correct or wrong:

$$\bar{e}^{\mathrm{cla}}(\Theta) = \frac{1}{M_{\mathrm{val}} \cdot L} \sum_{l=0}^{M_{\mathrm{val}} \cdot L - 1} e^{\mathrm{cla}}(l,\Theta). \tag{7.15}$$

It can be given in percent, where a value of 0% would indicate that no errors occur. The frame-wise decision for one of the classes is done based on the index of the highest value in the predicted output vector:

$$e^{\mathrm{cla}}(l,\Theta) = \begin{cases} 0 & \text{for } \operatorname{argmax}(\hat{\mathbf{y}}(l,\Theta)) = \operatorname{argmax}(\mathbf{y}(l)) \\ 1 & \text{otherwise.} \end{cases} \tag{7.16}$$

7.3 Input Feature Selection

This section deals with the results of the input feature selection which was discussed in section 6.2. The forward selection algorithm was used in order to find a feature set that achieves a high performance while maintaining the network dimensions low. As described above, two evaluations are presented in the following. The basic algorithm selects suitable features for an MSE-based ABE that predicts the WB envelope. These features are then compared to the features that are selected by a second algorithm that predicts whether a NB input belongs to a sibilant fricative or not. The comparison of the regression and the classification approach is finally discussed. The results that are presented in the following are based on a prior work related to this thesis [SFS18].

7.3.1 Training Setup

All DNNs that were trained for the feature selection approach had the same training setup. In order to achieve comparable results, just the size of the input feature vector was changed between two trainings. Pairs of sentences in WB and NB were generated according to section 7.1.2. The target feature was set to 30 MFCCs for all regression DNNs and to two classes (sibilant fricatives and other phonemes) for the classification DNNs. The model consisted of two hidden, fully connected feedforward layers with 64 nodes each, followed by a ReLU activation function. The mini-batch size was set to $L = 128$ frames, the initial learning rate to $\eta = 1 \cdot 10^{-5}$, and the L2 regularization weight to $\gamma^{\mathrm{reg}} = 2 \cdot 10^{-3}$. No auto-encoder was used to pre-train the variables in the network and the MSE was used as loss function.

The evaluation of the regression network is done based on the MSE of the MFCCs of predicted and true WB speech. Inserting the cepstral coefficients $c_{\mathrm{wb}}(i_c, l)$ as output features $y_i(l)$ in eq. (2.11) leads to the loss function

$$J^{\mathrm{mse}}(l, \Theta) = \sum_{i_c=0}^{N_c-1} |c_{\mathrm{wb}}(i_c, l) - \hat{c}_{\mathrm{wb}}(i_c, l, \Theta)|^2. \tag{7.17}$$

This loss function is evaluated on trained networks with different input feature sets in order to choose the best performing feature set. These feature sets are built by combining the feature set from the last iteration with one of the features from the feature pool. The averaged loss $\bar{J}^{\mathrm{mse}}(m, \Theta)$, which is the mean loss over all frames in a batch according to eq. (2.9), is hard to interpret. A more intuitive metric might be the MSE of the logarithmic mel-based PS described in eq. (7.5). Consequently, the averaged MSE over all M_{val} batches in the validation dataset

$$\bar{e}^{\mathrm{mse}\prime}(\Theta) = \frac{1}{M_{\mathrm{val}} \cdot L} \sum_{l=0}^{M_{\mathrm{val}} \cdot L-1} e_{\mathrm{wb}}^{\mathrm{mse}\prime}(l, \Theta) \tag{7.18}$$

is shown in table 7.2.

i_{fs}	Added feature	Feature dim.	Set dim.	$\bar{e}^{\text{mse}\prime}(\Theta)$	Rel. impr. [%]
1	mfc	30	30	49.76	
2	Δmfc	20	50	48.17	3.20
3	san	4	54	47.61	4.32
4	hce	1	55	47.19	5.16
5	off	4	59	46.34	6.87
30			114	43.99	11.6

Table 7.2: Results of the feature evaluation for the regression DNN [SFS18]. In each iteration i_{fs}, the feature that leads to the lowest MSE of the normalized MFCCs is selected. The selected feature is given with its dimension and with the dimension of the resulting feature set. The evaluation function $\bar{e}^{\text{mse}\prime}(\Theta)$ is the MSE of the mel-based PSs, averaged over the validation dataset. It is shown instead of the MSE of the normalized MFCCs because it is easier to interpret. 'Rel. impr.' denotes the relative improvement compared to the loss after the first iteration in percent.

The classification DNN was evaluated using the cross-entropy (see section 2.2.1) of the predicted and the labeled data. Its target feature was a one-hot encoded vector of dimension 2. A SoftMax activation function was applied to the last layer in order to give predictions between 0 and 1. The selection procedure was equal to that of the regression networks. Like in table 7.2, a metric that can be interpreted more intuitively is shown in table 7.3: the error rate of the binary prediction in percent (see eq. (7.15)).

7.3.2 Results

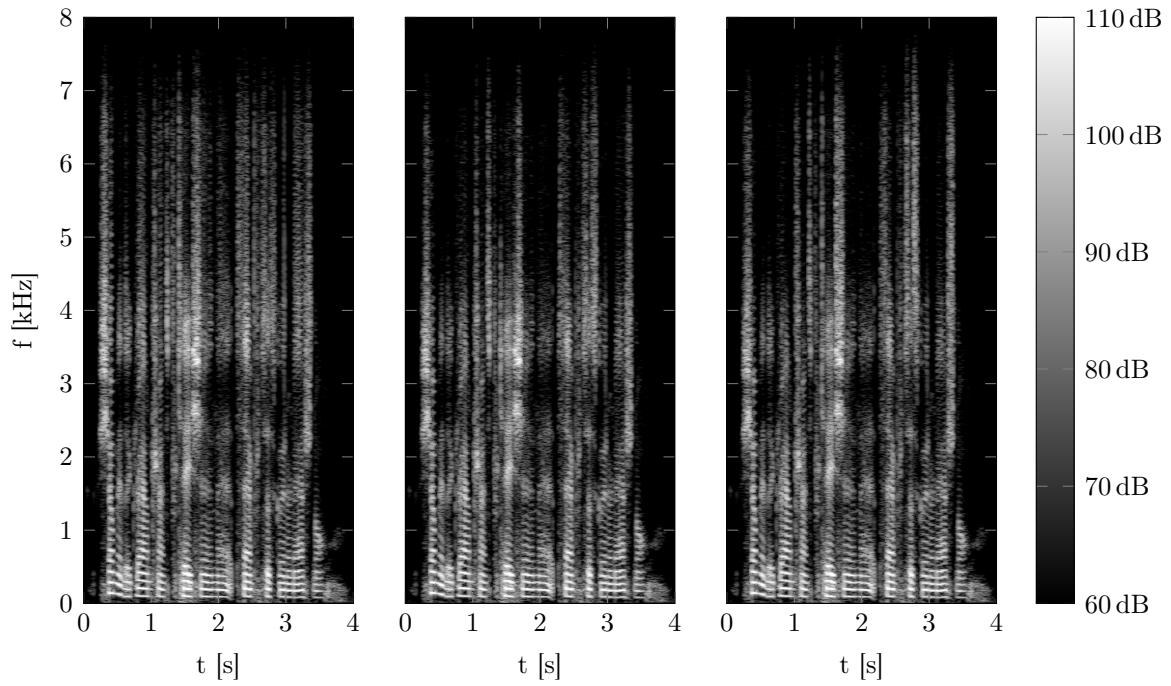
The results of the feature selection are shown for the regression DNN and the classification DNN in tables 7.2 and 7.3, respectively. The tables were filled row by row while the algorithm was running, starting at iteration index $i_{\text{fs}} = 1$. In each row, only the best-performing feature that was added to the set in the respective iteration is shown. Once added, the features remain in the set for the next iterations. The last line shows the results for the complete feature set that includes all features and their respective Δ- and $\Delta\Delta$-features.

The classification experiment in table 7.3 shows that the features which were selected in the regression approach were also suitable for the detection of sibilant fricatives. The order in which the features were added to the regression feature set is very similar to the resulting order of the classification task. This supports the decision of selecting the resulting features for an ABE application.

The resulting difference in the ABE algorithm can be regarded in fig. 7.2. Three spectrograms of the same NB speech signal after processing with ABE are shown. They show the benefit of using more input features while keeping the other network dimensions fixed. The feature selection algorithm for figs. 7.2a to 7.2c was stopped

i_{fs}	Added feature	Feature dim.	Set dim.	$\bar{e}^{\mathrm{cla}}(\Theta)$ [%]	Rel. impr. [%]
1	mfc	30	30	9.51	
2	Δmfc	20	50	7.57	20.4
3	$\Delta\Delta$mfc	10	60	7.15	24.8
4	hce	1	61	7.14	25.0
5	san	4	65	6.58	30.8
30			114	6.17	35.1

Table 7.3: Results of the feature evaluation for the classification of sibilant fricatives [SFS18]. The cross-entropy was used as selection criterion. $\bar{e}^{\mathrm{cla}}(\Theta)$ is the averaged rate of classification errors in percent for the whole validation dataset. It is shown here instead of the cross-entropy because it is easier to interpret. It should be noted that only about 10% of all phonemes belong to the class of sibilant fricatives. The false positive rate and the false negative rate was about equal for all results.



(a) 1 feature (b) 4 features (c) 30 features

Figure 7.2: Spectrograms of speech signals that were processed with ABE in dB. The WB envelope prediction is shown, which is based on only 30 MFCCs (a), on 4 features with an input vector length of 55 (b), and on the full feature vector of length 114 (c). The full feature vector contains all 10 features and their Δ- and $\Delta\Delta$-features. Brighter colors represent higher magnitudes. The true WB is not shown here as the speech signal is a recording from CDMA-coded NB speech that was not additionally recorded as 16 kHz WB signal.

after $i_{\text{fs}} = 1$, $i_{\text{fs}} = 4$ and $i_{\text{fs}} = 30$ iterations, respectively. The three versions mainly differ in their UB energy distribution over time. If the input feature set just consists of 30 MFCCs ($i_{\text{fs}} = 1$), most of the phonemes are extended similarly. Accordingly, the inserted UB energy is too high for vowels which leads to noisy artifacts and too low for sibilant fricatives. The DNN with the given model dimensions seems to need additional information to achieve a good separation between sibilant fricatives and other phonemes. Another interesting result is that the spectrogram of the extended speech signal using 4 features with a feature vector dimension of 55 already looks more similar to the one after all 30 iterations (dimension 114) than to the spectrogram based on 1 feature (30 MFCCs). This is another hint that the full feature set might not be the best trade-off between performance and complexity.

Generally, it seems that the FD features were preferred to the TD features by the selection algorithm. One reason for this might be that most of the FD features are more robust regarding different noise conditions than comparable TD features. This improved the performance in the given setup because the training data consisted of speech data in various noise conditions. The only drawback of the forward feature selection method is that the loss function that is used for the comparison does not always reflect subjective preferences.

7.4 Excitation Extension Methods

In chapter 5, two methods were proposed to extend the excitation spectrum from NB towards higher frequencies, namely MSS and MSSCN. Both methods are evaluated in this section. Subjective listening tests were conducted in [Sau+18a] as CCR tests in order to find differences in the excitation extension. Therein, MSS was compared to the classical excitation extension methods SF and SS and to the original WB excitation (OR). In a second comparison, it was investigated whether comfort noise insertion leads to an improvement by comparing MSSCN to MSS and OR. The speech samples that were used in the listening tests were synthesized by a multiplication of the extended excitation signal and the true WB envelope. This makes sure that the spectral envelope had no influence on the results. In the beginning of this chapter, the lack of objective metrics for ABE systems was mentioned. The same problem might hold for excitation extension algorithms. The quality of the algorithms was assessed in subjective listening tests instead of objective evaluations because of a lack of objective metrics.

7.4.1 Subjective Listening Test

CCR methods will offer a higher sensitivity than ACR methods if approximately equivalent conditions are rated [Möl12]. Consequently, all ratings were done as comparisons between two signals in a CCR test. The CMOS scale with values between -3 and 3 was used as rating scale (see table 7.1c). The following description of the listening test was adapted from [Sau+18a].

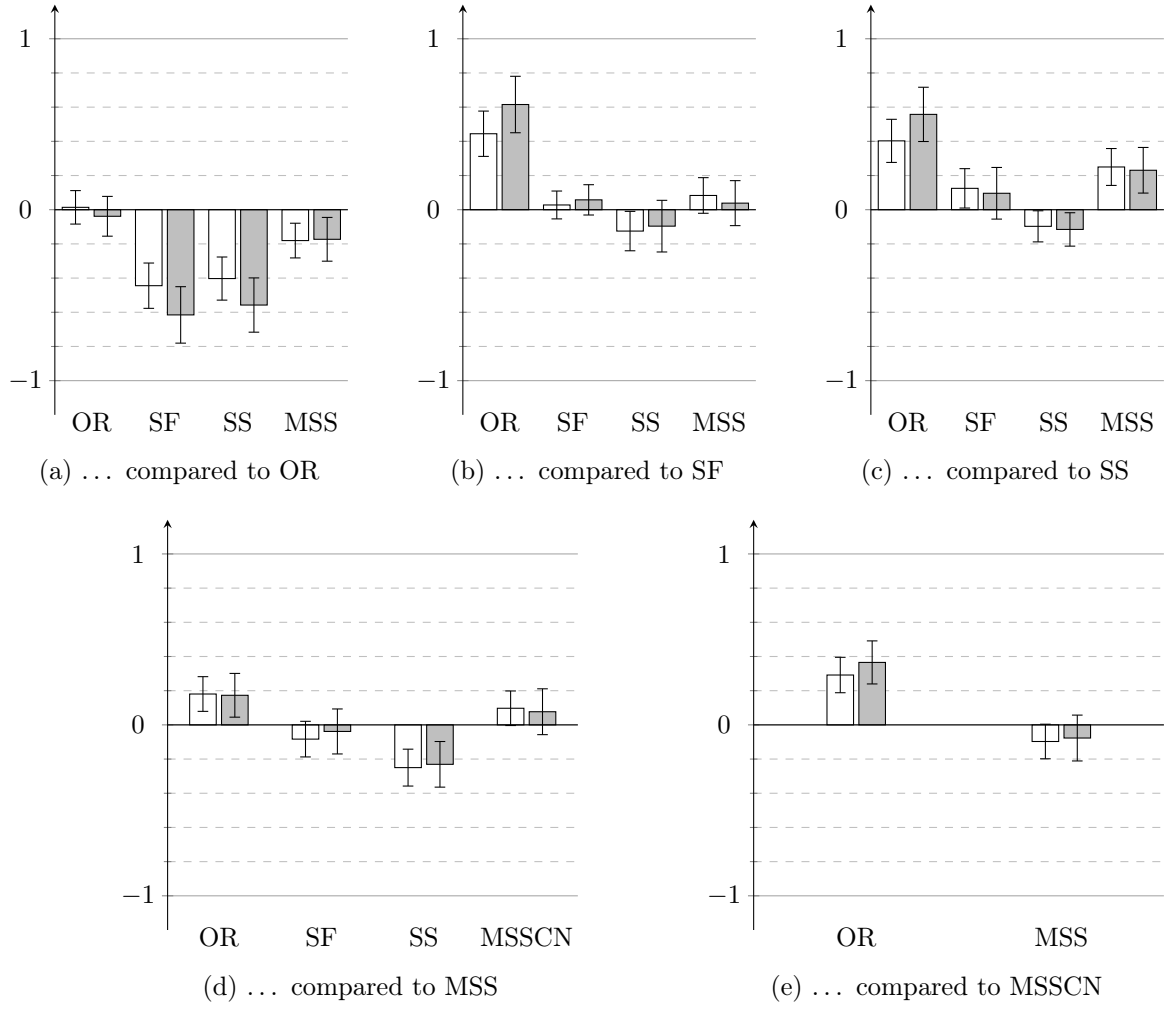
Data Generation For the evaluation, 100 utterances from the TIMIT corpus (see section 7.1.2) that were not included in the training data have been selected. The sentences were chosen randomly with the only restriction that 50 speakers had to be female and 50 male. All sound signals were normalized to a level of -5 dBFS. The true WB spectral envelope and the WB excitation were extracted from the WB speech signal. As a preparation for the excitation extension, the corresponding NB excitation was created by removing the parts above 4 kHz with a lowpass filter. Every NB excitation was extended to WB by SF, SS, MSS, and MSSCN. The extended excitations were finally multiplied with the original WB envelope to synthesize the predicted speech spectrum.

Test Setup A CMOS test was conducted to evaluate the overall speech quality. 36 listeners participated, of which 28 were male and 8 female. They were between 23 and 51 years old. Every participant was asked to rate the quality difference between two signals A and B in 22 comparisons. The speech utterances were selected randomly from the created dataset. The excitation classes OR, SF, SS, and MSS were compared with each other. In order to evaluate the effect of inserting comfort noise, MSSCN was compared to OR and MSS. As a reliability check, OR, SF, and SS were additionally compared with themselves. All these comparisons were repeated in reversed order of the examples A and B, which led to the $(3 + 2 + 1 + 2 + 3) \cdot 2 = 22$ comparisons per listener, which were presented in a randomized order. All sound signals could be played back multiple times. The volume of the playback could also be chosen by the listeners. A short introduction was given before the rating started, in which the participants could listen to 10 random A-B comparisons. This should help the listeners to get used to the size of the differences.

Results The results of the listening test are summarized in figs. 7.3a to 7.3e. The bars display the averaged ratings of the A-B comparisons. The length of the error bars denote the standard error which represents the 95% confidence intervals (see eq. (7.2)).

All listeners had a mean absolute rating smaller than 1 on the CMOS scale. This indicates that, on average, all differences between the excitation extension methods were rated less than *slightly better* or *slightly worse*. On some signals, the differences seem to be completely inaudible for most listeners. Prior informal listening tests further supported this hypothesis.

Figure 7.3a summarizes the results of all direct comparisons between OR and one of OR, SF, SS, and MSS. While SF and SS are rated slightly worse than OR, the distance between OR and MSS is much smaller. Regarding the confidence intervals, just the differences between the classical methods (SF and SS) and OR are statistically significant. No clear preference could be observed in the direct comparison between SF and SS (see fig. 7.3b). MSS was rated better than SS with statistical significance in fig. 7.3c but just marginally better than SF in fig. 7.3b. Comparing MSS and MSSCN did not result in a significant difference.



(a) ... compared to OR (b) ... compared to SF (c) ... compared to SS

(d) ... compared to MSS (e) ... compared to MSSCN

Figure 7.3: Mean rating $\bar{r}$ (bars) and standard error $\sigma_{\bar{r}}$ (error plots) of the subjective listening tests. The results for all listeners are shown as white bars, gray bars indicate the ratings of the reliable listeners only. A positive value indicates that the listeners prefer the respective method that is given on the x-axis to the reference method, which is OR in (a), SF in (b), and so on. Note that the limits of the y-axis represent the subjective levels of *slightly better/worse* and that the total range of the CMOS score is $\{-3, \ldots, 3\}$.

As stated in the last paragraph, OR, SF, and SS were also compared with themselves. These comparisons were conducted in order to be able to check the reliability of each listener in the evaluation of the MDVR (see eq. (7.4)). 10 out of 36 listeners had an MDVR larger than 1 which indicates that those listeners could not distinguish whether the signals they listened to were equal or different. Those listeners were categorized as being less reliable regarding the small quality differences and their ratings were excluded in the gray bars in fig. 7.3. The hypothesis that most of the differences were at the border of audibility is supported by the high amounts of about 40% of the non-experts and 23% of the experts who rated with $MDVR > 1$.

The results for reliable listeners (gray bars) are similar to the results of all listeners (white bars). All trends stay the same. The biggest difference might be that the quality difference between the extensions methods and OR is rated slightly larger by the reliable listeners. This indicates that the listeners who were rated as unreliable were also able to hear the small trends and that the differences were sometimes just too small to hear them. When taking into account the uncertainty regarding the comparison between twice OR, the confidence intervals of OR and MSS in fig. 7.3a overlap and the original signal was therefore not rated significantly better than MSS.

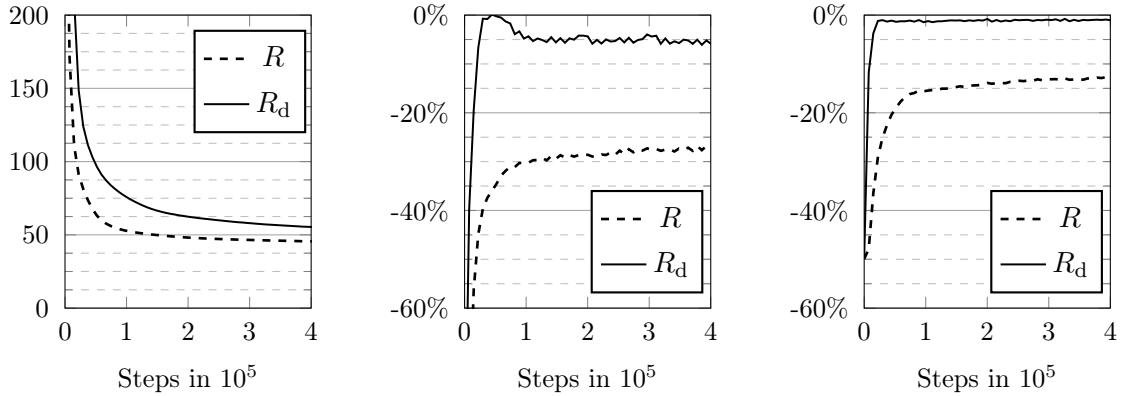
The insertion of comfort noise is evaluated in fig. 7.3e. No statistically significant improvements of the subjective quality could be observed in the comparison of MSSCN and MSS as the standard error was higher than the absolute mean rating score.

7.4.2 Discussion

Comparing the differences between OR and all extension methods, MSS is nearly as good as OR while SF and SS are a bit worse. Looking at all results compared to MSS, the difference to SF is not statistically significant. However, combining all the comparisons, MSS should be preferred to SF and SS because of its slightly better quality and the unchanged computational complexity when implemented in the FD. Further research might not be necessary as the quality difference between MSS and OR is just at the border of audibility with a CMOS score of 0.2. The insertion of noise towards higher frequencies did not yield statistically significant improvements of the speech quality and is therefore not necessary.

7.5 Discriminative Training

In this section, the results of the discriminative training approach that was explained in section 6.4 are presented and evaluated according to [Sau+18b]. The training setup is described in section 7.1. Objective quality measures are used in section 7.5.1 to show that the over-smoothing effect is reduced. The overall quality is rated in subjective listening tests in section 7.5.2. Finally, the results are discussed in section 7.5.3.



(a) $e_{\mathrm{ub}}^{\mathrm{mse}\prime}(m,\Theta)$ (b) $e_{\mathrm{ub}}^{\sigma}(m,\Theta)$ (c) $e^{q}(m,\Theta)$

Figure 7.4: Objective measures plotted while training the DNN by simple MSE regression (R, dotted curves) and in combination with discriminative training (R_d, solid curves). The metrics that are depicted in the plots are the MSE of the logarithmic mel-based UB PS $e_{\mathrm{ub}}^{\mathrm{mse}\prime}(m,\Theta)$ (a), the relative error of the UB standard deviation $e_{\mathrm{ub}}^{\sigma}(m,\Theta)$ (b), and the relative error of the SFPR (m,Θ) (c).

7.5.1 Objective Quality Measures

Three objective metrics were observed in the training process. The MSE of the logarithmic mel-based UB PS $e_{\mathrm{ub}}^{\mathrm{mse}\prime}(m,\Theta)$ (see eq. (7.5)) is shown as metric for the overall convergence in fig. 7.4a. A low value indicates that the distribution of the energy over time and frequency fits well to the original WB spectrum. As the distribution over time is generally more important, especially regarding different phoneme classes, this distribution is targeted in the other two metrics: The relative error of the UB standard deviation $e_{\mathrm{ub}}^{\sigma}(m,\Theta)$ (see eq. (7.12)) gives a hint whether the dynamics over time, which are based on the predicted speech signals, fit to the real WB speech. It is depicted in fig. 7.4b for the training process. The relative error of the SFPR, $e^{q}(m,\Theta)$ (see eq. (7.14)), focuses on the energy ratio between sibilant fricatives and other phonemes and is shown in fig. 7.4c.

The standard deviation of the UB power level is strongly underestimated by the MSE-based ABE system ($e_{\mathrm{ub}}^{\sigma}(m,\Theta^{R})$, dashed line). The value after convergence of about -27% can be adjusted to -7% by using a discriminative term in the loss function ($e_{\mathrm{ub}}^{\sigma}(m,\Theta^{R_\mathrm{d}})$, solid line) during the training of the DNN.

Adding the discriminative term was thought to force the network to reproduce the SFPR of true WB data in the prediction. This can be clearly observed in the SFPR. The final value for the MSE-based ABE system ($e^{q}(m,\Theta^{R})$, dashed line) is about 13% too low. This deviation is reduced by discriminative training to about -1% ($e^{q}(m,\Theta^{R_\mathrm{d}})$, solid line), which can be accepted as a final result.

In order to check whether the targets of the discriminative training and the regression task stand in conflict, the MSE of the true and the predicted logarithmic mel-based PSs

in the UB is evaluated. In the graph, the MSE is slightly higher for the discriminative approach which might be caused by the interfering targets. The weights of the two loss function parts γ^{mse} and γ^{dis} in eq. (6.21) were set to achieve a trade-off between a small deviation of the SFPR and a good convergence of $e_{\text{ub}}^{\text{mse}\prime}(m, \Theta)$.

7.5.2 Subjective Listening Tests

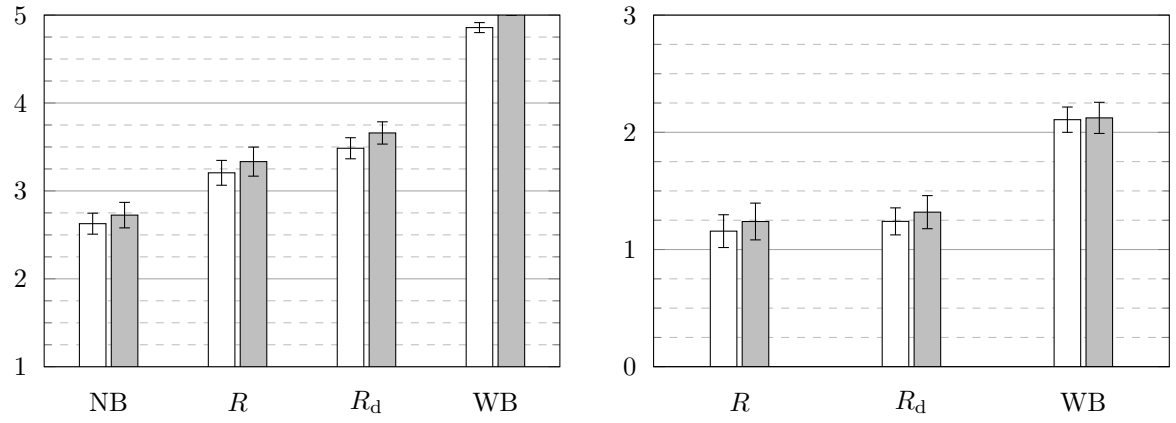
The speech quality of ABE with discriminative training was evaluated by subjective listening tests. In the following, the test data generation, the test setup, and the results are presented according to [Sau+18b].

Data Generation The test data consisted of six German sentences, recorded as 16 kHz WB data. Each sentence was read by a different speaker out of three male and three female speakers. By training on English speech and evaluating on German speech data, it was ensured that no language dependent advantages were exploited. The WB data was lowpass-filtered at a cutoff frequency of 4 kHz to generate the artificial NB data. Note that this is an idealized NB filter: it does not attenuate low frequencies as these are not in the focus of this thesis and the true cutoff filter that is defined individually by the cellphone manufacturers mostly lies somewhere between 3.4 and 3.9 kHz. Two ABE systems that are based on different DNN trainings were applied to the NB speech signals in order to create the WB signals. For both trainings, the excitation extension method of MSS was chosen. One of the trainings used a DNN, named R, that was just trained with the MSE loss function. A discriminative term was added to the MSE loss function for the second ABE training, resulting in the DNN R_{d}. The conditions are named like the DNNs in the following.

Listening Test Setup In a DMOS test (see section 7.2.1), the participants were asked to rate the perceived degradation of a test signal relative to the reference (WB). All conditions, NB, R, R_{d}, and WB, were used as test signals. The comparison between two identical WB signals was used to test the reliability of the listeners. All 6 speech signals were presented once to every listener which resulted in $6 \cdot 4 = 24$ comparisons. The order of the comparisons was randomized, while the reference signal (WB) was always the first one that was presented to the listeners.

In a second test, R, R_{d}, and WB were compared to NB to evaluate the perceived speech quality improvement of the ABE methods. This test was realized as a CMOS test (see section 7.2.1). The same six speech signals as in the DMOS test were presented once to every listener in a shuffled order of the examples A and B. The participants were not told that NB speech was always one of the categories A and B. The CMOS test included $6 \cdot 3 = 18$ additional comparisons for every listener. Like in the DMOS test, the comparisons were presented in a random order.

Randomly, the listeners had to start with the DMOS or the CMOS test. Altogether, both tests sum up to $24 + 18 = 42$ ratings per listener. 34 listeners participated in



(a) DMOS: ... compared to WB

(b) CMOS: ... compared to NB

Figure 7.5: Mean speech quality ratings (bars) and 95% confidence intervals c_{95} (error plots) of the DMOS test (a) and the CMOS test (b). The results are shown for all listeners (white) and selected listeners (gray). The y-axis in (b) is just shown for positive values as NB was rated worse than all compared categories.

the test, of which 25 were male and 9 female. 19 of them were working in the field of speech processing and will be called experts in the following. The average age of all participants was 31.

Listening Test Results The mean answers of all 34 listeners are shown as white bars in figs. 7.5a and 7.5b. The gray bars indicate the results for those 23 listeners who correctly heard no differences between equal sound signals. All other 11 participants were taken out of the calculation for the gray bars. Additionally, the 95% confidence intervals c_{95} are depicted as error plots. The reliability of the listeners is evaluated based on the WB-to-WB comparisons in the DMOS test.

In the results of the reliable participants, it can be seen that both ABE methods yield high improvements compared to NB with CMOS ratings of 1.24 for R and 1.32 for R_d. The difference between both ABE methods is higher in the DMOS test in fig. 7.5a, where R_d is rated better than R, and where the 95% confidence intervals do not overlap ($c_{95}(R) = \{3.17, \ldots, 3.50\}$, $c_{95}(R_\mathrm{d}) = \{3.53, \ldots, 3.79\}$). These relations persist similarly in case the results of the unreliable listeners are also taken into account in the white bars.

However, in the CMOS test, the 95% confidence intervals overlap and the mean values are only slightly different. The reason for this might be that the difference in the bandwidth is the most dominant part in the comparisons between NB and ABE processed speech. The minor artifacts introduced by different ABE algorithms seem to be clearer when comparing with WB speech where the bandwidth is similar. However, a direct comparison between R and R_d was not tested.

7.5.3 Conclusion

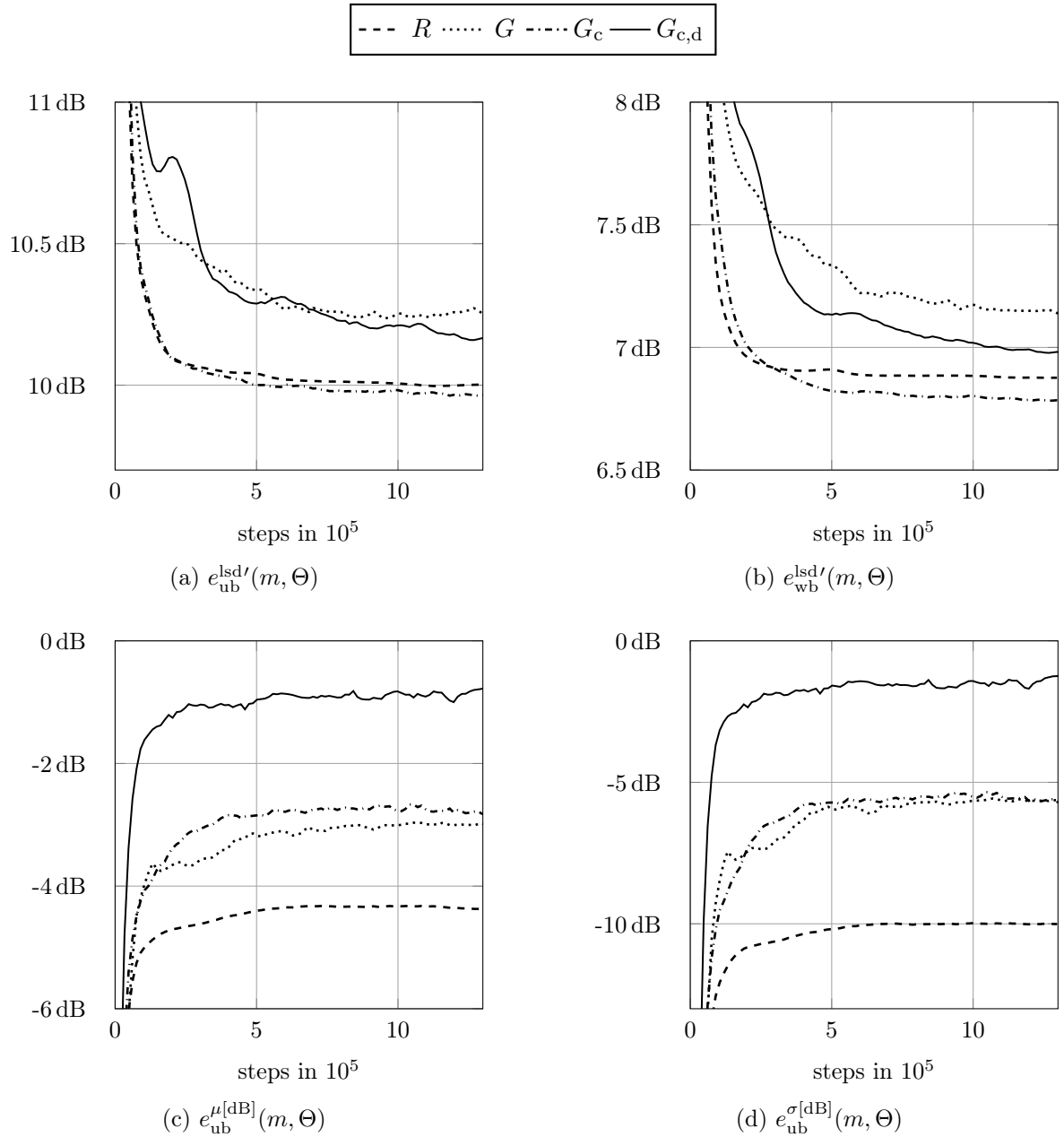
In this section, the problem of over-smoothing in the training of a regression DNN for ABE was addressed. Objective measures showed how discriminative training effectively reduces the deviation of the SFPR. In a CMOS listening test, it was verified that the proposed ABE method improved the quality of the NB speech by about 1.3 points on the CMOS scale. The results of a DMOS listening test showed that the ABE with discriminative training yields better results than the ABE that is just based on the MSE loss function. Discriminative training improved the perceived quality slightly by 0.33 points on the DMOS scale.

7.6 Combination with Adversarial Training

The method of adversarial training was introduced in section 6.5. In this section, the results will be presented following [Sau+19]. The training setup is described in section 7.1. It differs slightly from the training setup of the discriminative training comparison in section 7.5. This is caused by the parameter fine-tuning that was done between both evaluations which led to slightly improved results. Objective quality measures are used in section 7.6.1 to show that the desired effect of reducing the over-smoothing while maintaining a good overall convergence is achieved in the training. Because of the lack of objective quality measures for ABE [Bau+14; Pul+15; Möl+13], the overall quality is rated in subjective listening tests in section 7.6.2. Finally, the results are discussed in section 7.6.3.

7.6.1 Objective Quality Measures

Some effects that lead to a low perceived speech signal quality were collected from informal feedback of the participants of the listening tests. These are similar to the main challenges for ABE that were formulated in previous studies: A first effect that was often mentioned is an undesired lisping sound. This mainly occurs when the introduced UB energy for sibilant fricatives like [s] and [z] is not high enough. A second effect that can lead to a perceived quality degradation is the insertion of high UB energy where it should be low, especially for vowels or breathing noise. Both effects have in common that an evenly distributed energy over time that does not consider different phoneme classes degrades the perceived speech quality. The energy distribution over time is assessed in the following by inspecting the mean $\mu_{\mathrm{ub}}^{[\mathrm{dB}]}(m)$ and the standard deviation $\sigma_{\mathrm{ub}}^{[\mathrm{dB}]}(m)$ of the frame-wise UB power over time $p'_{\mathrm{ub}}(m)$ [Sau+18b]. The deviation of these measures between the true and the predicted WB data is plotted in figs. 7.6c and 7.6d. As expected, the negative values show that the mean and the variance of the predicted UB power levels are too low. The underestimation of those measures gives a hint for the strength of the over-smoothing effect regarding the basic DNN R. While the generative networks, G in the GAN and G_{c} in the CGAN, lead to a



(a) $e_{\text{ub}}^{\text{lsd}\prime}(m,\Theta)$ (b) $e_{\text{wb}}^{\text{lsd}\prime}(m,\Theta)$

(c) $e_{\text{ub}}^{\mu[\text{dB}]}(m,\Theta)$ (d) $e_{\text{ub}}^{\sigma[\text{dB}]}(m,\Theta)$

Figure 7.6: Performance measures of the training process of four different DNN models (R, G, G_c, and $G_{c,d}$). The training steps are given on the x-axes in units of 10^5 and the unit of the y-axis is dB for all graphs. The first two plots show the LSD of the UB mel spectrum (a) and the WB mel spectrum (b). In the last two plots, the predicted and the true values are compared for the mean and the variance of the UB power level $p'_{\text{ub}}(l,\Theta)$. (c) shows the deviation of the mean $e_{\text{ub}}^{\mu[\text{dB}]}(m,\Theta)$ and (d) the deviation of the standard deviation $e_{\text{ub}}^{\sigma[\text{dB}]}(m,\Theta)$.

similar improvement regarding mean and variance, the deviations are strongly reduced by discriminative training in the CGAND (CGAN with discriminative training) $G_{\mathrm{c,d}}$.

The metrics presented above focus on the statistical distribution of the UB energy per time frame. They do not take into account whether the right frames are extended with a high energy. Furthermore, they are independent of the energy distribution along the frequency axis. To monitor these aspects, LSD measures of the UB or the entire WB mel spectrum can be used (see eqs. (7.6) and (7.7)).

The baseline DNN R underestimates the UB dynamics but yields low distance measures. While the energy distribution over time can be improved by using a GAN in G, the distance measures increase. The CGAN G_{c} combines the lower distance measure that the regression DNN R achieved while maintaining the low distribution deviations $|e_{\mathrm{ub}}^{\mu[\mathrm{dB}]}(m,\Theta)|$ and $|e_{\mathrm{ub}}^{\sigma[\mathrm{dB}]}(m,\Theta)|$. Discriminative training is, like GAN training, a step towards a better fit of the distribution to the cost of a higher LSD. A trade-off was found between correct distribution and low spectral distance in order to maximize the perceived speech quality in $G_{\mathrm{c,d}}$.

7.6.2 Subjective Listening Tests

Subjective listening tests were conducted to evaluate the perceived speech quality of the presented ABE methods. The following paragraphs describe the listening test setup and its results.

Listening Test Setup The speech data consisted of 8 German sentences, half of them spoken by a female and the other half of a male speaker. The following 6 conditions were tested in a CMOS test:

- **NB**: artificially created by filtering a WB signal with a steep lowpass at 4 kHz
- R: ABE using a regression DNN applied to NB
- G: ABE using the generator of a modified GAN applied to NB
- G_{c}: ABE using the generator of a CGAN applied to NB
- $G_{\mathrm{c,d}}$: ABE using the generator of a CGAND applied to NB
- **WB**: clean speech with a bandwidth of up to 8 kHz

All ABE approaches were applied to artificially generated NB signals. For each spoken sentence, the proposed method from [Sau+19], $G_{\mathrm{c,d}}$, was compared to all other conditions once. In every comparison, the quality of a condition A was rated relative to a condition B. The order of the comparisons and of the examples A and B was randomized. The reliability of the listeners was tested by additionally presenting the same sample of $G_{\mathrm{c,d}}$ as both A and B in a test once for each sound signal. These 6 comparisons per sound signal sum up to a total of 48 comparisons per participant. 24 listeners (21 male and 3 female), who were between 25 and 59 years old, participated in

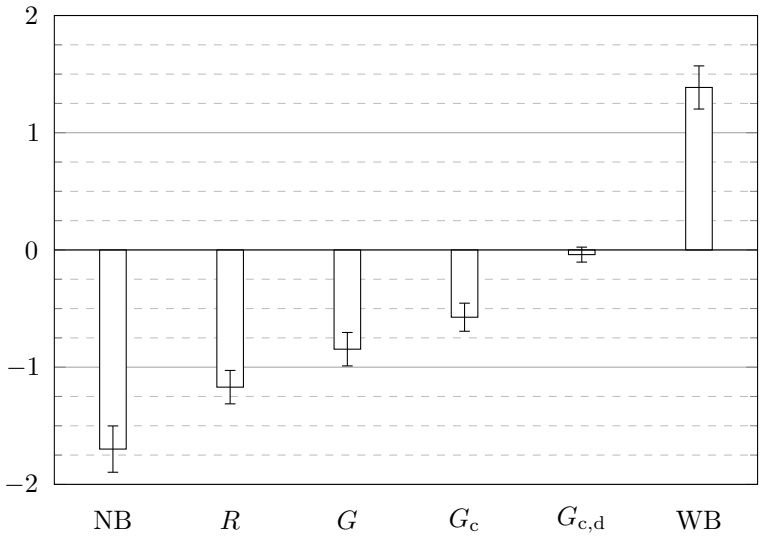


Figure 7.7: Mean CMOS ratings (bars), compared to $G_{\mathrm{c,d}}$, and 95% confidence intervals (error plots). The limits of the y-axis do not show the whole range of possible CMOS values which is $\{-3, \ldots, 3\}$. The mean and standard error at the comparison of two $G_{\mathrm{c,d}}$ examples are not exactly zero as some of the listeners heard differences that are depicted in the bar plot.

the test. 21 of them were experts in the field of speech processing. All participants were German speakers. This is especially important for ABE because the listeners have to know how the words and phonemes should be pronounced. A lisping effect that leads to a strong quality reduction can only be realized correctly by a listener who is familiar with the sharp articulation of [s] and [z] in the German language.

Listening Test Results The results of the listening test are presented in fig. 7.7 as CMOS values. The rank order supports the hypothesis that all the modifications resulted in an improvement of the perceived speech quality. These results are statistically significant because there is no overlap of the 95% confidence intervals. Especially the high quality improvement of 1.7 CMOS points from NB to $G_{\mathrm{c,d}}$ is remarkable. This perceived quality difference exceeds the difference between $G_{\mathrm{c,d}}$ and WB (1.39 CMOS points). A possible conclusion is that more than 50% on the way from NB quality to WB quality are already reached by the $G_{\mathrm{c,d}}$ approach. The improvement compared to NB is also way higher than the improvement that was achieved using a basic DNN with discriminative training R_{d} in the listening tests that focused on discriminative training (1.24 CMOS points in section 7.5). However, it has to be noted that some of the general training parameters changed between both evaluations that could also be one of the reasons for the good quality.

A common feedback that was given by listeners was that one had to decide whether a higher speech bandwidth or slight artifacts were preferable. Especially some of the experts rated the negative impact of introduced artifacts much higher than the advantage

of a higher bandwidth. If this tendency also holds for a bigger sample, a study without experts might even have shown a higher quality improvement between NB and ABE.

In a small pre-study, the effects of applying discriminative training and replacing the GAN by a CGAN seemed to be independent of each other so that they might sum up. The objective results support this hypothesis because the discriminative training aims at optimizing the statistical distribution over time while the CGAN training is thought to minimize the distance measure also regarding the spectral distribution. As a result, the combination of both approaches in a CGAND training yielded a superior quality.

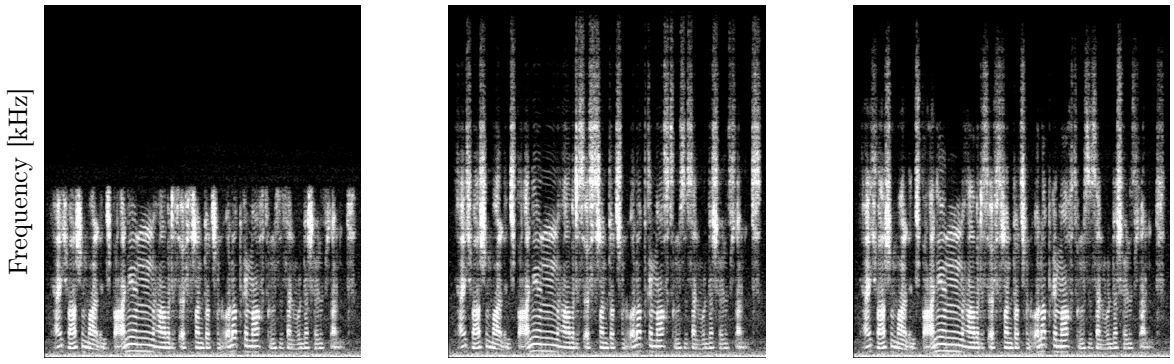
Another approach that also uses a GAN for ABE was already proposed by Li et al. in 2018 [Li+18]. The main differences to the current work are that the GAN architecture was replaced by a CGAN here and that discriminative training was applied. These modifications showed a significant improvement of 0.8 points on the CMOS scale in the performance when comparing the CGAND ($G_{\mathrm{c,d}}$) with the pure GAN approach (G).

7.6.3 Discussion

Training a pure GAN for ABE would train the discriminator independently of the NB input features. These features can be used in a CGAN because they are feeded to the discriminator as additional input. The additional input in the CGANs yielded a significant improvement regarding the speech quality in subjective listening tests. Although the quality was improved with CGANs, the over-smoothing problem could not completely be solved. Hence, it was proposed to combine the CGAN architecture with discriminative training in order to combine the high quality and the correct energy relations between different phoneme classes in the UB. The subjective listening tests proved the superior speech quality of the combined ABE approach that was rated 1.7 CMOS points better than NB.

7.7 ABE in Simulated Driving Situation

All former studies in this thesis were conducted in a laboratory situation. The listeners were able to concentrate on small differences because of the quiet environment and the noise-free speech samples. This situation is completely different from the scenario of driving in a car while listening to another person's voice through loudspeakers. It is interesting to find the limits of ABE in theory but in the end the practical aspects are more important for the listener in the car. The listening test described in this section is intended to focus more on the real in-car scenario. The acoustic environment in the car was simulated including driving noise in order to find out whether this has an effect on the subjective rating of ABE algorithms. The listening test and its results are presented in the first part of this section in section 7.7.1. The interpretation of the results is done in section 7.7.2.



(a) NB (b) $G_{\mathrm{c,d}}$ (c) WB

Figure 7.8: NB, ABE, and WB spectra for the left channel of a binaural signal that was used in the listening test.

7.7.1 Subjective Listening Tests

The aim of this listening test was to prove whether the quality improvement introduced by the proposed ABE algorithm from section 6.5 also holds for a more realistic environment. In order to be able to interpret the ratings, the NB and ABE-processed signals were also compared to the WB signal. The other listening tests in this thesis were based on clean 16 kHz speech recordings as WB signals and a lowpass-filtered version as NB signals. In contrast to this, the incoming signal of a real implementation of ABE is also modified by the transmission over the telephone network. This means that the signal gets encoded and decoded and that the distant cellphone might influence the bandpass characteristics. The general setup of the listening test and the processing that was done in order to simulate the real driving situation are described in the following paragraphs. Afterwards, the results are presented.

Listening Test Setup A CMOS test was chosen to evaluate the differences between AMR-NB, ABE-processed AMR-NB, and AMR-WB. It is important to note that the AMR-WB signals had the same HP characteristics in low frequencies as the AMR-NB signals. This was done to just evaluate the changes in the UB as the ABE was just applied towards higher frequencies. The ABE condition was generated using the generator $G_{\mathrm{c,d}}$ of the CGAND network from section 7.6. All three conditions, NB, $G_{\mathrm{c,d}}$, and WB, were compared with each other, and comparisons of two identical speech signals were used to check the reliability of the listeners. Each listener had to rate four comparisons per speech signal (all three combinations with different conditions and one randomly chosen comparison of two identical signals). The speech recordings that were used in this listening test stem from the Speecon database [Isk+02]. Six male and six female speakers were randomly selected from the recordings of spontaneous speech in German, each reading one sentence of about 4–5 seconds. Spontaneous speech was selected because it is a common use-case and might occur more often than read speech

in phone calls. For every listener, three male and three female speakers were selected, which results in $4 \cdot 6 = 24$ comparisons per listener. Because of the binaural synthesis, the speech signals had to be played back through headphones. 32 persons participated in the listening test (22 male and 10 female), of which 14 were classified as experienced listeners.

Speech Data Processing The following processing steps were applied to the speech signals in order to simulate the situation of driving in a car while listening to the distant speaker in a phone call:

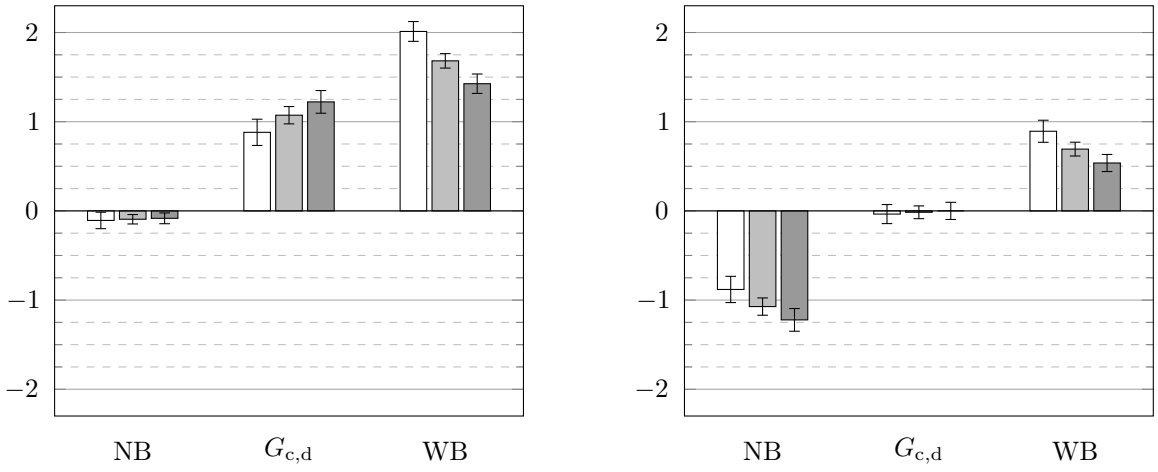
- **Noise Suppression**
  The low-level background noise that was found in some recordings was removed by a stationary noise suppression.

- **Narrowband Filter**
  The characteristics of the bandpass filter that is applied to a NB speech signal before it is sent to the network differs highly for different cellphones. In the 1970s, the send characteristics of many analogue telephones was measured and an average filter was derived, which was called intermediate reference system (IRS) [ITU88b]. A modified version was later proposed for digital handsets under the name modified intermediate reference system (MIRS) [ITU96b]. The MIRS characteristics was applied to the speech recordings to generate NB speech.

- **Wideband Filter**
  The WB characteristics of cellphones is mostly rather flat between 200 Hz and 7 kHz. A filter that was proposed in [ITU11a] for the hands-free sending sensitivity was chosen as WB filter. This is also recommended in [ETS18] for the simulation of a handset in order to assess the speech quality of a WB speech signal. For the filter from ITU recommendation P.341 and for the MIRS filter, the reference implementations from the ITU were used [ITU19]. As this thesis just focuses on the bandwidth extension towards higher frequencies, there should not be any difference in the lower frequencies between the NB, ABE-processed, and WB signals. To achieve the same HP characteristics, the lower part of the MIRS filter up to 2 kHz was also applied to the WB signal.

- **Lowpass**
  The MIRS filter attenuates the signal slightly towards higher frequencies. In order to remove all contents above 4 kHz in the NB signal, an additional LP with a cutoff frequency of 4 kHz was applied to the signal.

- **Codecs**
  The NB signals were transcoded with AMR-NB at 12.2 kbps and the WB signals with AMR-WB at 12.65 kbps.



(a) ... compared to NB (b) ... compared to $G_{c,d}$

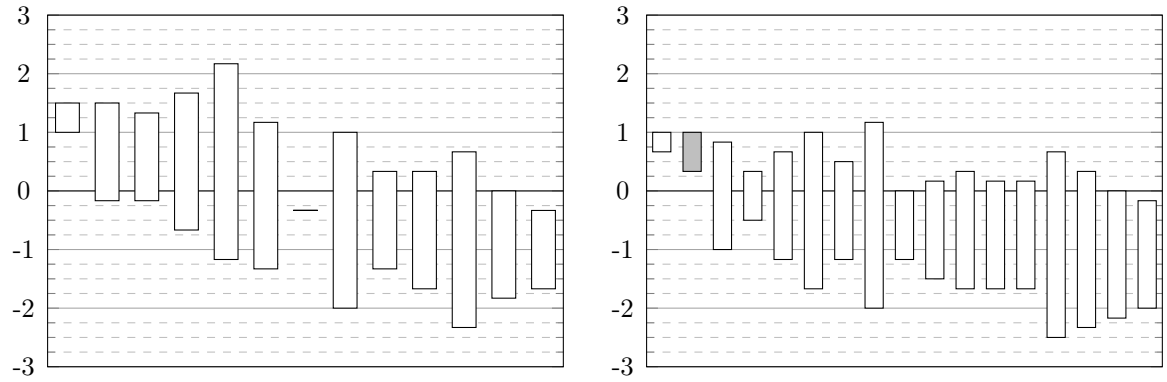
Figure 7.9: Mean CMOS ratings (bars) and 95% confidence intervals c_{95} (error plots) of the comparison with NB (a) and with $G_{c,d}$ (b). The white bars represent the results of the expert listeners, the light gray bars the results of all listeners, and the dark gray bars the results of the non-experts. The limits of the y-axis do not represent the whole range of possible CMOS values which is $\{-3, \ldots, 3\}$.

- **Binaural Simulation**
  For a more realistic listening experience, the acoustic environment of the car was simulated. Impulse responses from the front loudspeakers in a car to the ears of an artificial head, sitting on the driver's seat, had already been measured. These impulse responses were convolved with the NB and WB signals. Each channel in the resulting stereo sound signal was the sum of the simulated speech signals originating from the left and the right loudspeaker in the car.

- **Background Noise**
  Stationary background noise was recorded in the same car with the microphones of the artificial head at a speed of about 100 km/h. It was mixed with the speech signals for both channels at a fixed SNR of 10 dB.

- **Normalization**
  All signals were finally normalized to the same root mean square (RMS) level, which was just calculated on those samples where the smoothed absolute amplitude exceeded a threshold. This threshold guaranteed that the amount of silence in the signals did not have an influence on the volume of the speech parts.

Listening Test Results The results of the listening test in fig. 7.9 show a clear preference of the ABE-processed speech signals compared to AMR-NB speech of nearly 1.1 CMOS points. The quality difference between $G_{c,d}$ and WB is just 0.7 on the CMOS scale. This means that the quality gap between AMR-NB and AMR-WB could be closed



(a) Experts (b) Non-experts

Figure 7.10: Listener-based evaluation of $G_{\mathrm{c,d}}$ compared to NB and WB for expert (a) and non-expert (b) listeners. Each bar represents one listener with the NB-to-$G_{\mathrm{c,d}}$ rating as lower end and the WB-to-$G_{\mathrm{c,d}}$ rating as upper end of the bar. Only for the single gray bar, the limits are swapped because in comparison with $G_{\mathrm{c,d}}$, the listener rated NB better than WB. The horizontal line at a value of zero represents the quality of $G_{\mathrm{c,d}}$.

by 60% regarding the extension towards higher frequencies. The percentage changes to 70% when only non-expert ratings are evaluated and to 50% when only experts are asked. In order to visualize the difference between experts and non-experts as well as the high individual differences, the comparison of $G_{\mathrm{c,d}}$ to both other conditions is shown for every listener in fig. 7.10. Each bar represents one listener. The lower end of the bar denotes the quality of NB relative to $G_{\mathrm{c,d}}$ and the upper end denotes the quality of WB relative to $G_{\mathrm{c,d}}$. The horizontal line at zero shows the ratio with which the quality gap is closed for each listener. The bars are sorted by their mean height in descending order for experts in fig. 7.10a and for non-experts in fig. 7.10b.

7.7.2 Discussion

A CMOS test showed that the gap between AMR-NB and AMR-WB could be closed by 60% regarding the extension towards higher frequencies. When the extension towards lower frequencies is also taken into account, the rating for the ABE methods might even be higher. However, as also the rating for AMR-WB might increase, the change of the percentage with which the quality gap is closed cannot be predicted.

It is remarkable that the ratings of the experienced listeners deviate so strongly from the ratings of the other listeners. It seems that the unexperienced listeners mainly rated the higher bandwidth as a positive effect. The experienced listeners were better in finding artifacts introduced by the ABE algorithms or they at least put a higher emphasis on these artifacts. However, it was shown that the subjective rating varied a lot also in the groups of expert and non-expert listeners.

Chapter 8

Conclusion and Outlook

8.1 Conclusion

ABE has been a research topic for many years now. But still, no perfect solution has been found that would solve the problem completely. Since DNNs have become a very powerful tool to solve mapping problems in the last years, new opportunities came up also for ABE. The aim of this thesis is to find an ABE algorithm that is based on neural networks and that outperforms earlier approaches. A special requirement is a high cost-efficiency which allows for a real-time implementation on a relatively low performing processor in a car.

In this thesis and the related work, several methods have been implemented that extend the bandwidth of a NB speech signal to WB. In all methods, the excitation and the envelope of the NB signal are extended separately. The excitation extension methods do not use neural networks because the results of simple shifting approaches already showed a sufficient performance. The method that performed best in the comparisons was called multiple spectral shifting (MSS). It shifts a part from the middle of the NB spectrum multiple times up to higher frequencies in order to extend the bandwidth. As the participants in a subjective listening test could not clearly distinguish between MSS and the original WB excitation, there might not be the need for further improvements.

For the extension of the spectral envelope, MFCCs of the WB spectral envelope are predicted in a DNN. The WB coefficients are estimated from a combination of the NB MFCCs and several additional features. In order to find a set of input features for this task that is at the same time small and performant, a feature selection algorithm was applied. The resulting feature sets that were used in the ABE approaches of this work contain up to 20 features with a total input feature dimension of up to 137. A separate feature selection approach on a classification task showed that these features are also suitable for detecting sibilant fricatives in a speech signal. This ability is very important for an ABE algorithm as a wrong detection of sibilant fricatives can lead to a high degradation of the quality of the extended signal.

The estimation of the UB energy distribution is the most difficult task in ABE. In the scope of this thesis, DNNs were trained in different ways and they were compared in

subjective listening tests. The comparisons show that a pure feedforward DNN, trained with the MSE as loss function, is not able to predict the WB target accurately. The main problem in the MSE training is the over-smoothing effect, which means for ABE that the DNN does not distinguish strongly enough between phoneme classes with a different amount of energy in the UB. This effect is reduced partly by discriminative training, which means that a change in the loss function forces the network to reproduce differences between the phoneme classes. Another reduction of over-smoothing has been achieved by training the network as part of a CGAN in an adversarial training approach. The combination of both methods led to the best performing network for ABE. In a final listening test that simulated the scenario of having a phone call in the car while driving, its superior quality was proved once again.

8.2 Outlook

Neural networks are a big topic in current research and nearly every year new training methods and architectures are published. Many of these enhancements might also improve the quality of an ABE algorithm. However, only a subset of the huge amount of possibilities could be investigated in the scope of this thesis. In this outlook, some of the possibilities that seem to have a high potential are briefly explained.

Recurrent Networks Recurrent networks are neural networks that incorporate additional dependencies in the time dimension. They are used frequently in time-series problems, in which the current time frame depends on the preceding frames. It is obvious that this property is given for human speech in which many phonemes have a duration of multiple frames. Well known examples are the LSTMs (long-short term memorys) and the GRUs (gated recurrent units). A speech activity detection can also be made more robust when having the possibility to learn time dependencies. A combination with the discriminative loss function and the adversarial training could improve the results further.

Convolutional Networks Convolutional neural networks (CNNs) use convolutions in order to simplify problems using a high number of parameters. They are often applied to extract patterns in image recognition tasks and similar topics. In speech processing, CNNs make it possible to take the TD signal as input to the network. The high dimension of the input vector can be efficiently reduced and patterns in the waveform can be extracted. The advantage of this approach is that no information can be lost before it is given to the network and no feature extraction is necessary. However, it might not be the best approach for a small network with low dimensions as it is the aim in this thesis.

There are many more possibilities to perform ABE with neural networks and the approaches presented in this thesis are not meant to give an exhaustive overview of the topic.

Appendix A

Abbreviations and Notation

List of Abbreviations

ABE	artificial bandwidth extension
ACR	absolute category rating
AMR	adaptive multi rate
ANN	artificial neural network
BAM	bidirectional associative memory
BP	bandpass
BP-MGN	bandpass-envelope modulated Gaussian noise
CCR	comparison category rating
CDMA	code-division multiple access
CELP	code excited linear prediction
CGAN	conditional GAN
CGAND	CGAN with discriminative training
CMOS	comparison mean opinion score
CNN	convolutional neural network
CQT	constant Q transform
dB	decibel, logarithmic unit for level of signal power
dBFS	decibel full scale, level in dB compared to amplitude of 1
DCR	degradation category rating

DCT	discrete cosine transform
DFT	discrete Fourier transform
DIAL	diagnostic instrumental assessment of listening quality
DMOS	degradation mean opinion score
DNN	deep neural network
EQ	equalizer
ETSI	European Telecommunications Standards Institute
EVRC	enhanced variable rate codec
FB	fullband
FD	frequency domain
FFT	fast Fourier transform
FIR	finite impulse response
GAN	generative adversarial network
GD	gradient descent
GMM	Gaussian mixture model
GRU	gated recurrent unit
GSM	Global System for Mobile Communications
HMM	hidden Markov model
HNM	harmonic noise model
HNR	harmonics-to-noise ratio
HP	highpass
IDCT	inverse discrete cosine transform
IDFT	inverse discrete Fourier transform
IIR	infinite impulse response
IoT	internet of things
IPA	international phonetic alphabet
IR	impulse response
IRS	intermediate reference system

ISTFT	inverse short-term Fourier transform
ITU	International Telecommunication Union
LEM	loudspeaker-enclosure-microphone
LP	lowpass
LPC	linear predictive coding
LSD	log-spectral distance
LSF	line spectral frequency
LSP	line spectral pair
LSTM	long-short term memory
MDVR	mean difference voting ratio
MFCC	mel frequency cepstral coefficient
MIRS	modified intermediate reference system
ML	machine learning
MLP	multilayer perceptron
MOS	mean opinion score
MSE	mean square error
MSS	multiple spectral shifting
MSSCN	multiple spectral shifting with comfort noise
MTL	multi-task learning
NB	narrowband
NE	noise estimation
NS	noise suppression
OR	original excitation
PCM	pulse code modulation
PESQ	perceptual evaluation of speech quality
POLQA	perceptual objective listening quality analysis
PReLU	parametric rectified linear unit
PS	power spectrum

RBM	restricted Boltzmann machine
ReLU	rectified linear unit
RIR	room impulse response
RMS	root mean square
RNN	recurrent neural network
RTRBM	recurrent temporal restricted Boltzmann machine
SAN	signal above noise
SD	speech distortion
SF	spectral folding
SFPR	sibilant fricative power ratio
SGD	stochastic gradient descent
SNR	signal-to-noise ratio
SS	spectral shifting
STFT	short-term Fourier transform
SWB	super-wideband
TD	time domain
UB	upper band
VAD	voice activity detection
WB	wideband

Notation

$\mathcal{N}(\mu, \sigma^2)$	normal distribution of mean μ and variance σ^2
$\mathcal{U}(a, b)$	uniform distribution with the limits a and b
$*$	convolution operator
d	derivative operator
∂	partial derivative operator
$\forall$	for all
$\Im$	imaginary part
∇	gradient operator

$\Re$	real part
Σ	sum operator
$\tilde{(\cdot)}$	combination of true and predicted values
$\hat{(\cdot)}$	prediction
$\bar{(\cdot)}$	mean value over time frames or over ratings of a listening test
$\lvert\cdot\rvert$	absolute value
$\lVert\cdot\rVert_2^2$	squared euclidean norm, L2-norm
$(\cdot)'$	variable is based on the mel-scale
$(\cdot)^*$	complex conjugate
$(\cdot)^{\Delta}$	delta feature, first derivative of a feature over time
$(\cdot)^{\Delta\Delta}$	delta-delta feature, second derivative of a feature over time
$(\cdot)^{\mathsf{T}}$	transpose
$\mathcal{STFT}\{\cdot\}$	short-term Fourier transform

List of Latin Symbols

$a^{[\lambda]}(i)$	activation at the output of DNN layer λ at index i
$\mathbf{a}^{[\lambda]}$	activation vector at the output of DNN layer λ
$b^{[\lambda]}(i)$	bias of DNN layer λ at index i
$\mathbf{b}^{[\lambda]}$	bias vector of DNN layer λ
$c(i_c, l)$	mel-frequency cepstral coefficient at index i_c and frame l
c_{95}	95% confidence intervals
d	binary detection of a speech feature
e	Euler's constant
$e(m, \Theta)$	error measure at mini-batch m for the DNN Θ
$e^{\mu}(m, \Theta)$	relative error of the mean of the power level at mini-batch m for the DNN Θ
$e^{q}(m, \Theta)$	relative error of the SFPR at mini-batch m for the DNN Θ
$e^{\sigma}(m, \Theta)$	relative error of the standard deviation of the power level at mini-batch m for the DNN Θ

f	continuous frequency
f_0	fundamental frequency or pitch frequency of voiced speech
f_c	cutoff frequency of a HP/BP/LP filter
f_{Δ,n_h}	delta frequency of the harmonics with index n_h between two adjacent time frames
f_m	modulation frequency, size of the frequency shift
f_s	sample rate
$g^{[\lambda]}$	activation function of DNN layer λ
$h(n)$	impulse response or FIR filter
i	node index in the current layer of a DNN
$\imath$	alternative node index in the current layer of a DNN
$i^{[\lambda]}$	node index in layer λ of a DNN
i_c	index of a mel-frequency cepstral coefficient
i_fs	iteration index of the feature selection algorithm
i_r	index of a rating in a subjective listening test
j	imaginary unit
k	frequency band index
k_Δ	bandwidth of the shifted part in MSS in frequency bins
l	frame index
l_m	frame index in a mini-batch
m	mini-batch index
n	sample index
n_h	index of the harmonics of a sine wave
$p(l)$	mean power of a spectrum at frame l
$q(m)$	sibilant fricative power ratio (SFPR) at batch m
$r(i_r)$	rating score at rating index i_r
$\mathbf{r}$	rating score vector
$r_\mathrm{AA}(i_r)$	rating score for A-A comparisons at rating index i_r
$\mathbf{r}_\mathrm{AA}$	rating score vector for A-A comparisons

$r_{\mathrm{AB}}(i_r)$	rating score for A-B comparisons at rating index i_r
$\mathbf{r}_{\mathrm{AB}}$	rating score vector for A-B comparisons
$s(n)$	discrete speech signal
t	continuous time
u	arbitrary positive integer number
$v^{[\lambda]}(i)$	weighted sum of all inputs and the bias in DNN layer λ at index i
$\mathbf{v}^{[\lambda]}$	weighted sum of all input vectors and the bias vector in DNN layer λ
$w^{[\lambda]}(i^{[\lambda]}, i^{[\lambda-1]})$	DNN weight between node $i^{[\lambda]}$ in layer λ and node $i^{[\lambda-1]}$ in layer $\lambda - 1$
$w_{\mathrm{ana}}(n)$	analysis window
$w_{\mathrm{syn}}(n)$	synthesis window
$x(i, l)$	input of DNN at index i and frame l
$\mathbf{x}(l)$	input feature vector of DNN at frame l
$y(i, l)$	output of DNN at index i and frame l
$\mathbf{y}(l)$	output feature vector of DNN at frame l
$z(n)$	discrete acoustic noise signal
$\mathbf{z}(l)$	noise input to GAN at frame l
C	classification network
D	discriminator network in GAN
D_{c}	discriminator network in conditional GAN
F	length of frame shift in samples for frame-based processing
G	generator network in GAN
G_{c}	generator network in conditional GAN
$G_{\mathrm{c,d}}$	generator network in conditional GAN with discriminative training
$J(l, \Theta)$	loss function based on frame l and the network parameters Θ
K	number of frequency bands

L	number of frames in a mini-batch (mini-batch size)
L_{tot}	total number of frames in a spectrogram
$\mathcal{L}$	number of layers in a DNN
M	number of mini-batches in the training data
$\mathcal{M}(k', k)$	mel transformation factor from frequency index k to mel-band index k'
M_{val}	number of mini-batches in the validation data
N	number of samples in a frame, DFT length
N_{av}	length of sliding average window
N_c	number of mel-frequency cepstral coefficients
$N^{[\lambda]}$	number of nodes in DNN layer λ
$N_{\mathbf{r}}$	number of ratings in a subjective listening test
$N_{\mathbf{r}_{\text{AA}}}$	number of A-A ratings in a subjective listening test
$N_{\mathbf{r}_{\text{AB}}}$	number of A-B ratings in a subjective listening test where A and B are different
$N_{\mathbf{x}}$	length of input feature vector $\mathbf{x}$
$N_{\mathbf{y}}$	length of target feature vector $\mathbf{y}$
R	regression network
R_{d}	regression network with discriminative training
$S(k, l)$	short-term spectrum of $s(n)$
$\mathbf{W}^{[\lambda]}$	DNN weight matrix vector in layer λ
$\mathbf{X}(m)$	batch of DNN input feature vectors at batch index m
$\mathbf{Y}(m)$	batch of DNN output feature vectors of at batch index m
$Z(k, l)$	short-term spectrum of white noise

List of Greek Symbols

α	factor in leaky or parametric ReLU activation function
$\beta(k)$	frequency dependent weight for comfort noise interpolation
γ	weight for loss function part
$\delta(i)$	Kronecker delta, is 1 for $i = 0$ and 0 otherwise

η	learning rate or step size for DNN training
λ	layer index
$\lambda_{\sin}$	wavelength of a sine wave
μ	mean value
$\mu_{\mathbf{x}}$	mean value of the input features
$\mu_{\mathbf{y}}$	mean value of the target features
π	constant pi
ρ	exponent for comfort noise interpolation
σ	standard deviation
$\sigma_{\mathbf{x}}$	standard deviation of the input features
$\sigma_{\bar{r}}$	standard error of r
$\sigma_{\mathbf{y}}$	standard deviation of the target features
σ^2	variance
θ	threshold value for the detection of a speech feature
$\psi(n)$	normed gradient of a signal
ω_m	modulation frequency
$\Theta(m)$	variables of a DNN at batch index m
$\Phi_{SS}(k,l)$	power spectrum of $S(k,l)$
$\Psi(n)$	indicator for a change of the normed gradient

Indices

$(\cdot)^{\text{ce}}$	cross-entropy loss function part
$(\cdot)^{\text{cgan}}$	CGAN loss function part
$(\cdot)^{\text{cla}}$	error rate of a classification DNN
$(\cdot)^{[\text{dB}]}$	logarithmic value in dB
$(\cdot)^{\text{dis}}$	discriminative loss function part
$(\cdot)^{\text{env}}$	spectral envelope of the given spectrum
$(\cdot)^{\text{exc}}$	excitation signal of the given spectrum
$(\cdot)^{\text{gan}}$	GAN loss function part

$(\cdot)^{[\lambda]}$	variable belongs to layer λ of a DNN
$(\cdot)^{\text{lsd}}$	LSD of the PS at frame l
$(\cdot)^{\text{mse}}$	MSE loss function part
$(\cdot)^{\text{reg}}$	regularization loss function part
$(\cdot)_{\text{bp}}$	bandpass
$(\cdot)_{\text{cen}}$	variable refers to the centroid feature
$(\cdot)_{\text{gri}}$	variable refers to the gradient-index feature
$(\cdot)_{\text{hce}}$	variable refers to the high centroid feature
$(\cdot)_{\text{hi}}$	upper end of the frequency range that is copied in MSS
$(\cdot)_{\text{hp}}$	highpass
$(\cdot)_{\text{kur}}$	variable refers to the kurtosis feature
$(\cdot)_{\text{lev}}$	variable refers to the power level feature
$(\cdot)_{\text{lo}}$	lower end of the frequency range that is copied in MSS
$(\cdot)_{\text{lp}}$	lowpass
$(\cdot)_{\text{mfc}}$	variable refers to the MFCC feature
$(\cdot)_{\text{mss}}$	excitation extension was done using multiple spectral shifting
$(\cdot)_{\text{msscn}}$	excitation extension was done using MSS with comfort noise
$(\cdot)_{\text{nb}}$	narrowband version of the given signal or spectrum
$(\cdot)_{\text{off}}$	variable refers to the offset feature
$(\cdot)_{\text{ons}}$	variable refers to the onset feature
$(\cdot)_{\text{san}}$	variable refers to the signal-above-noise feature
$(\cdot)_{\text{sf}}$	excitation extension was done using spectral folding
$(\cdot)_{\text{sfr}}$	variable refers to the phoneme class of sibilant fricatives
$(\cdot)_{\text{ss}}$	excitation extension was done using spectral shifting
$(\cdot)_{\text{ub}}$	upper band version of the given signal or spectrum
$(\cdot)_{\text{wb}}$	wideband version of the given signal or spectrum
$(\cdot)_{\text{zcr}}$	variable refers to the zero crossing rate feature

List of Figures

2.1 Graph of an MLP. 11
2.2 Graph of an artificial neuron in an MLP. 12
2.3 Training and validation loss over time with overfitting in the training. 13
2.4 Frequently used activation functions for DNNs: sigmoid functions and rectifiers. 18
2.5 Schematic graph of the pre-training of a DNN with a stacked auto-encoder. 23
2.6 Graphs with the structural differences between GAN and CGAN training. 25

3.1 Anatomic scheme and abstract model of the vocal tract. 29
3.2 Subdivision of different speech sounds. 30
3.3 Comparison of the mean NB and the mean UB energy of phonemes in the TIMIT dataset in dB. 31
3.4 Mean spectra of different phoneme classes. 32
3.5 Mean energy of different phonemes in the NB and the UB for the TIMIT dataset. 32
3.6 Source-filter model of speech production. 33
3.7 Schematic example of the source-filter model of speech production for a voiced speech sound. 34
3.8 Attenuation limits for circuits with 4-kHz channel equipment, adapted from ITU-T rec. G.120. 35
3.9 Attenuation limits for the hands-free sending path of cellphones, adapted from ITU-T rec. P.341. 35
3.10 Analysis and synthesis window $w_{\text{ana}}(n)$ that is used in the filterbank. . 37
3.11 Raw mel filterbank without normalization. 38

4.1 General block diagram of ABE in the FD, based on the source-filter model of speech production. 43
4.2 Spectrum of the vowel [æ] in the word 'favor'. 43
4.3 Schematic spectra for the three excitation extension methods: SF, SS, and MSS. 45
4.4 WB and NB short-term spectral envelopes of the phonemes [æ] and [s]. 47

4.5 Block diagram of the training of a DNN and the prediction of a WB envelope using the trained DNN. 48

5.1 Spectrogram of a speech excitation signal for WB, NB, and for the extension methods SF, SS, MSS, and MSSCN. 54

6.1 Clean and noisy NB spectrogram of a speech signal from the TIMIT database with a comparison of several features. 66

7.1 Block diagram of the training data generation process for a DNN that predicts the WB spectral envelope from features that are based on the NB signal. 77
7.2 Spectrograms of speech signals that were processed with ABE with a differing number of DNN input features. 86
7.3 Mean rating $\bar{r}$ and standard error $\sigma_{\bar{r}}$ of the subjective listening tests. . 89
7.4 Objective measures plotted while training the DNN by simple MSE regression and in combination with discriminative training. 91
7.5 Mean speech quality ratings and 95% confidence intervals of the DMOS test and the CMOS test. 93
7.6 Performance measures of the training process of four different DNN models (R, G, G_{c}, and $G_{\mathrm{c,d}}$). 95
7.7 Mean CMOS ratings, compared to $G_{\mathrm{c,d}}$, and 95% confidence intervals . 97
7.8 NB, ABE, and WB spectra for the left channel of a binaural signal that was used in the listening test. 99
7.9 Mean CMOS ratings of simulated driving scenario evaluation and 95% confidence intervals c_{95}. 101
7.10 Listener-based evaluation of $G_{\mathrm{c,d}}$ compared to NB and WB for expert and non-expert listeners. 102

List of Tables

3.1 Modes of the NB codecs. 40
3.2 Bitrates of WB codecs. 40

6.1 Pool of input features for a DNN with TD and FD features. 62

7.1 Rating scales for subjective listening tests according to ITU-T recommendation P.800. 80
7.2 Results of the feature evaluation for the regression DNN. 85
7.3 Results of the feature evaluation for the classification of sibilant fricatives. 86

Bibliography

[Abe+16] J. Abel et al. "A subjective listening test of six different artificial bandwidth extension approaches in English, Chinese, German, and Korean". In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Shanghai, China: IEEE, Mar. 2016, pp. 5915–5919.

[Abe+17] Johannes Abel et al. "An instrumental quality measure for artificially bandwidth-extended speech signals". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 25.2 (Feb. 2017), pp. 384–396.

[AF17] Johannes Abel and Tim Fingscheidt. "A DNN regression approach to speech enhancement by artificial bandwidth extension". In: *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, Oct. 2017, pp. 219–223.

[AF18] Johannes Abel and Tim Fingscheidt. "Artificial speech bandwidth extension using deep neural networks for wideband spectral envelope estimation". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 26.1 (Jan. 2018), pp. 71–83.

[AHW95] C. Avendano, H. Hermansky, and E. A. Wan. "Beyond Nyquist: towards the recovery of broad-bandwidth speech from narrow-bandwidth speech." In: *1995 European Conference on Speech Communication and Technology (EUROSPEECH)*. 1995.

[Bac+17] Pramod B. Bachhav et al. "Artificial bandwidth extension using the constant Q transform". In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, Mar. 2017, pp. 5550–5554.

[BAF14] Patrick Bauer, Johannes Abel, and Tim Fingscheidt. "HMM-based artificial bandwidth extension supported by neural networks". In: *2014 International Workshop on Acoustic Signal Enhancement*. IEEE, Sept. 2014, pp. 1–5.

[Bat94] R. Battiti. "Using mutual information for selecting features in supervised neural net learning". In: *IEEE Transactions on Neural Networks* 5.4 (July 1994), pp. 537–550.

[Bau+14] P. Bauer et al. "On speech quality assessment of artificial bandwidth extension". In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Florence, Italy: IEEE, May 2014, pp. 6082–6086.

[BB12] James Bergstra and Yoshua Bengio. "Random search for hyper-parameter optimization". In: *Journal of Machine Learning Research* 13.10 (2012), pp. 281–305.

[Bes+02] B. Bessette et al. "The adaptive multirate wideband speech codec (AMR-WB)". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 10.8 (Nov. 2002), pp. 620–636.

[BF09] P. Bauer and T. Fingscheidt. "A statistical framework for artificial bandwidth extension exploiting speech waveform and phonetic transcription". In: *2009 European Signal Processing Conference (EUSIPCO)*. Glasgow, Scotland, Aug. 2009, pp. 1839–1843.

[Car97] Rich Caruana. "Multitask learning". In: *Machine Learning* 28.1 (1997), pp. 41–75.

[CH94] H. Carl and U. Heute. "Bandwidth enhancement of narrow-band speech signals". In: *1994 European Signal Processing Conference (EUSIPCO)*. Vol. 2. 1994.

[CH96] Cheung-Fat Chan and Wai-Kwong Hui. "Wideband re-synthesis of narrowband CELP-coded speech using multiband excitation model". In: *1996 International Conference on Spoken Language Processing (ICSLP)*. Vol. 1. IEEE, Oct. 1996, pp. 322–325.

[COM92] Yan Ming Cheng, D. O'Shaughnessy, and P. Mermelstein. "Statistical recovery of wideband speech from narrowband speech". In: *1992 International Conference on Spoken Language Processing (ICSLP)* (Oct. 1992), pp. 1577–1580.

[COM94] Yan Ming Cheng, D. O'Shaughnessy, and P. Mermelstein. "Statistical recovery of wideband speech from narrowband speech". In: *IEEE Transactions on Speech and Audio Processing* 2.4 (Oct. 1994), pp. 544–548.

[Cro72] M. G. Croll. *Sound-quality improvement of broadcast telephone calls*. Research rep. 26. British Broadcasting Corporation, Jan. 1972.

[DLP17] Chris Donahue, Bo Li, and Rohit Prabhavalkar. "Exploring speech enhancement with generative adversarial networks for robust speech recognition". In: *CoRR* (Nov. 15, 2017).

[EH99] J. Epps and W. H. Holmes. "A new technique for wideband enhancement of coded narrowband speech". In: *1999 IEEE Workshop on Speech Coding Proceedings. Model, Coders, and Error Criteria*. IEEE, 1999, pp. 174–176.

[EK99] N. Enbom and W. B. Kleijn. "Bandwidth expansion of speech based on vector quantization of the mel frequency cepstral coefficients". In: *1999 IEEE Workshop on Speech Coding*. 1999.

[Erh+10] D. Erhan et al. "Why does unsupervised pre-training help deep learning?" In: *Journal of Machine Learning Research* 11.2 (2010), pp. 625–660.

[ETS18] ETSI Technical Report 103 138. *Speech and multimedia transmission quality (STQ); Speech samples and their use for QoS testing*. Tech. rep. 103 138, V1.5.1. ETSI, Aug. 2018.

[Fan60] C. G. M. Fant. *Acoustic theory of speech production: With calculations based on x-ray studies of russian articulations*. Description and analysis of contemporary standard russian. Mouton, 1960.

[FHG01] J. A. Fuemmeler, R. C. Hardie, and W. R. Gardner. "Techniques for the regeneration of wideband speech from narrowband speech". In: *EURASIP Journal on Advances in Signal Processing* 2001.1 (Jan. 2001), pp. 266–274.

[Gar+93] J. S. Garofolo et al. *DARPA TIMIT acoustic phonetic continuous speech corpus CDROM*. 1993.

[GB10] X. Glorot and Y. Bengio. "Understanding the difficulty of training deep feedforward neural networks". In: *2010 International Conference on Artificial Intelligence and Statistics (AISTATS)*. 2010, pp. 249–256.

[GBB11] X. Glorot, A. Bordes, and Y. Bengio. "Deep sparse rectifier neural networks". In: *2011 International Conference on Artificial Intelligence and Statistics (AISTATS)*. Ed. by Geoffrey Gordon, David Dunson, and Miroslav Dudík. Vol. 15. Proceedings of Machine Learning Research. Fort Lauderdale, FL, USA: PMLR, Apr. 2011, pp. 315–323.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. `http://www.deeplearningbook.org`. MIT Press, 2016.

[GL15] Yu Gu and Zhen-Hua Ling. "Restoring high frequency spectral envelopes using neural networks for speech bandwidth extension". In: *2015 International Joint Conference on Neural Networks (IJCNN)*. Killarney, Ireland: IEEE, July 2015, pp. 1–8.

[GL17] Y. Gu and Z. H. Ling. "Waveform modeling using stacked dilated convolutional neural networks for speech bandwidth extension". In: *2017 Conference of the International Speech Communication Association (INTERSPEECH)*. Aug. 2017, pp. 1123–1127.

[GLD16] Y. Gu, Z. H. Ling, and L. R. Dai. "Speech bandwidth extension using bottleneck features and deep recurrent neural networks". In: *2016 Conference of the International Speech Communication Association (INTERSPEECH)*. 2016.

[Goo+14] I. J. Goodfellow et al. "Generative adversarial nets". In: *2014 International Conference on Neural Information Processing Systems (NIPS) - Volume 2*. NIPS'14. Montreal, Canada: MIT Press, 2014, pp. 2672–2680.

[He+15] Kaiming He et al. "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification". In: *CoRR* (Feb. 6, 2015).

[HKA05] R. Hu, V. Krishnan, and D. Anderson. "Speech bandwidth extension by improved codebook mapping towards increased phonetic classification". In: *2005 Conference of the International Speech Communication Association (INTERSPEECH)*. Jan. 2005, pp. 1501–1504.

[IMS08] B. Iser, W. Minker, and G. Schmidt. *Bandwidth extension of speech signals*. Ed. by Bernd Iser, Wolfgang Minker, and Gerhard Schmidt. Springer Publishing Company, Incorporated, 2008.

[Int99] International Phonetic Association. *Handbook of the international phonetic association*. Cambridge University Press, June 28, 1999. 213 pp.

[IS03] B. Iser and G. Schmidt. "Neural networks versus codebooks in an application for bandwidth extension of speech signals". In: *2003 Conference of the International Speech Communication Association (INTERSPEECH)*. 2003.

[Isk+02] Dorota Iskra et al. "SPEECON - Speech databases for consumer devices: Database specification and validation". In: *2002 Conference on Language Resources and Evaluation (LREC)* (June 2002).

[Iso+16] Phillip Isola et al. "Image-to-image translation with conditional adversarial networks". In: *CoRR* (Nov. 21, 2016).

[ITU01] ITU-T Recommendation G.712. *Transmission performance characteristics of pulse code modulation channels*. Tech. rep. G.712. ITU, Nov. 2001.

[ITU03] ITU-T Recommendation G.722.2. *Wideband coding of speech at around 16 kbit/s using adaptive multi-rate wideband (AMR-WB)*. Tech. rep. G.722.2. ITU, July 2003.

[ITU05] ITU-T Recommendation P.862.2. *Wideband extension to recommendation P.862 for the assessment of wideband telephone networks and speech codecs*. Tech. rep. P.862.2. ITU, Nov. 2005.

[ITU11a] ITU-T Recommendation P.341. *Transmission characteristics for wideband digital loudspeaking and hands-free telephony terminals*. Tech. rep. P.341. ITU, Mar. 2011.

[ITU11b] ITU-T Recommendation P.863. *Perceptual objective listening quality assessment*. Tech. rep. P.863. ITU, Jan. 2011.

[ITU12] ITU-T Recommendation G.722. *7 kHz audio-coding within 64 kbit/s*. Tech. rep. G.722. ITU, Sept. 2012.

[ITU19] ITU-T Recommendation G.191. *Software tools for speech and audio coding standardization*. Tech. rep. G.191. ITU, Jan. 2019.

[ITU88a] ITU-T Recommendation G.711. *Pulse code modulation (PCM) of voice frequencies*. Tech. rep. G.711. ITU, Nov. 1988.

[ITU88b] ITU-T Recommendation P.48. *Specification for an intermediate reference system*. Tech. rep. P.48. ITU, Nov. 1988.

[ITU96a] ITU-T Recommendation P.800. *Methods for subjective determination of transmission quality*. Tech. rep. P.800. ITU, Aug. 1996.

[ITU96b] ITU-T Recommendation P.830. *Subjective performance assessment of telephone-band and wideband digital codecs*. Tech. rep. P.830. ITU, Feb. 1996.

[ITU98] ITU-T Recommendation G.120. *Transmission characteristics of national networks*. Tech. rep. G.120. ITU, Dec. 1998.

[Jär00] K. Järvinen. "Standardisation of the adaptive multi-rate codec". In: *2000 European Signal Processing Conference (EUSIPCO)*. Sept. 2000, pp. 1–4.

[Jax02] Peter Jax. "Enhancement of bandlimited speech signals: algorithms and theoretical bounds". Zsfassung in dt. u. engl. Sprache; Zugl.: Aachen, Techn. Hochsch., Diss., 2002. PhD thesis. Aachen: RWTH Aachen University, 2002.

[JV00] P. Jax and P. Vary. "Wideband extension of telephone speech using a hidden Markov model". In: *2000 IEEE Workshop on Speech Coding*. 2000.

[JV03a] P. Jax and P. Vary. "Artificial bandwidth extension of speech signals using MMSE estimation based on a hidden Markov model". In: *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Vol. 1. IEEE, Apr. 2003, pp. 680–683.

[JV03b] Peter Jax and Peter Vary. "On artificial bandwidth extension of telephone speech". In: *Signal Processing* 83.8 (Aug. 2003), pp. 1707–1719.

[JV04] P. Jax and P. Vary. "Feature selection for improved bandwidth extension of speech signals". In: *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Vol. 1. Montreal, Quebec, Canada: IEEE, May 2004, pp. 697–700.

[Kan+17] Takuhiro Kaneko et al. "Generative adversarial network-based postfilter for statistical parametric speech synthesis". In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, Mar. 2017, pp. 4910–4914.

[KB14] Diederik P. Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *CoRR* (Dec. 22, 2014).

[KJ97] Ron Kohavi and George H. John. "Wrappers for feature subset selection". In: *Artificial Intelligence* 97.1-2 (Dec. 1997). Relevance, pp. 273–324.

[KLA07] Juho Kontio, Laura Laaksonen, and Paavo Alku. "Neural network-based artificial bandwidth expansion of speech". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 15.3 (Mar. 2007), pp. 873–881.

[Kor01] U. Kornagel. "Spectral widening of the excitation signal for telephone-band speech enhancement". In: *2001 IEEE International Workshop on Acoustic Signal Enhancement (IWAENC)*. 2001, pp. 215–218.

[Led+16] Christian Ledig et al. "Photo-realistic single image super-resolution using a generative adversarial network". In: *CoRR* abs/1609.04802 (Sept. 15, 2016).

[Lee+18] Bong-Ki Lee et al. "Sequential deep neural networks ensemble for speech bandwidth extension". In: *IEEE Access* 6 (2018), pp. 27039–27047.

[Li+18] Sen Li et al. "Speech bandwidth extension using generative adversarial networks". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, Apr. 2018, pp. 5029–5033.

[Lin+18] Zhen-Hua Ling et al. "Waveform modeling and generation using hierarchical recurrent neural networks for speech bandwidth extension". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 26.5 (May 2018), pp. 883–894.

[LK16] Yaxing Li and Sangwon Kang. "Artificial bandwidth extension using deep neural network-based spectral envelope estimation and enhanced excitation estimation". In: *IET Signal Processing* 10.4 (June 2016), pp. 422–427.

[LL15] Kehuang Li and Chin-Hui Lee. "A deep neural network approach to speech bandwidth expansion". In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, Apr. 2015, pp. 4395–4399.

[LM96] P. Ladefoged and I. Maddieson. *The sounds of the world's languages*. Phonological Theory. John Wiley & Sons, Jan. 17, 1996. 450 pp.

[MB79] J. Makhoul and M. Berouti. "High-frequency regeneration in speech coding systems". In: *1979 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Vol. 4. Apr. 1979.

[MHN13] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. "Rectifier nonlinearities improve neural network acoustic models". In: *2013 International Conference on Machine Learning (ICML)*. Vol. 30. 1. 2013, p. 3.

[MO14] Mehdi Mirza and Simon Osindero. "Conditional generative adversarial nets". In: *CoRR* (Nov. 6, 2014).

[Möl+13] S. Möller et al. "Speech quality prediction for artificial bandwidth extension algorithms". In: *2013 Conference of the International Speech Communication Association (INTERSPEECH)*. 2013.

[Möl12] S. Möller. *Assessment and prediction of speech quality in telecommunications*. Springer Science & Business Media, 2012.

[NTN97] Y. Nakatoh, M. Tsushima, and T. Norimatsu. "Generation of broadband speech from narrowband speech using piecewise linear mapping." In: *1997 European Conference on Speech Communication and Technology (EUROSPEECH)*. 1997.

[PA11] H. Pulakka and P. Alku. "Bandwidth extension of telephone speech using a neural network and a filter bank implementation for highband mel spectrum". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 19.7 (Sept. 2011), pp. 2170–2183.

[Pat+16] D. Pathak et al. "Context encoders: Feature learning by inpainting". In: *CVPR 2016* (Apr. 25, 2016).

[Pat83] Peter J. Patrick. "Enhancement of band-limited speech signals". Ph.D. thesis. Loughborough University, Jan. 1983.

[PBS17] Santiago Pascual, Antonio Bonafonte, and Joan Serrà. "SEGAN: Speech enhancement generative adversarial network". In: *CoRR* (Mar. 28, 2017).

[PG03] Shahla Parveen and Phil Green. "Multitask learning in connectionist robust ASR using recurrent neural networks." In: *2003 European Conference on Speech Communication and Technology (EUROSPEECH)*. Jan. 2003.

[PK00] K.-Y. Park and H. S. Kim. "Narrowband to wideband conversion of speech using GMM based transformation". In: *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Vol. 3. IEEE, 2000, pp. 1843–1846.

[Pul+15] H. Pulakka et al. "Speech quality evaluation of artificial bandwidth extension: Comparing subjective judgments and instrumental predictions". In: *2015 Conference of the International Speech Communication Association (INTERSPEECH)*. Sept. 2015.

[QK03] Y. Qian and P. Kabal. "Dual-mode wideband speech recovery from narrowband speech". In: *2003 European Conference on Speech Communication and Technology (EUROSPEECH)*. 2003.

[RHW86a] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. "Parallel distributed processing: Explorations in the microstructure of cognition". In: *Parallel distributed processing: Explorations in the microstructure of cognition, Vol. 1: Foundations*. Cambridge, MA, USA: MIT Press, 1986. Chap. Learning Internal Representations by Error Propagation, pp. 318–362.

[RHW86b] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. "Learning representations by back-propagating errors". In: *Nature* 323 (1986), pp. 533–536.

[RM51] Herbert Robbins and Sutton Monro. "A stochastic approximation method". In: *The Annals of Mathematical Statistics* 22.3 (Sept. 1951), pp. 400–407.

[Ros62] F. Rosenblatt. *Principles of neurodynamics: Perceptrons and the theory of brain mechanisms*. Jan. 1962.

[RSS10] G. Ravindran, S. Shenbagadevi, and V. Salai Selvam. "Cepstral and linear prediction techniques for improving intelligibility and audibility of impaired speech". In: *Journal of Biomedical Science and Engineering* 03.01 (2010), pp. 85–94.

[Sau+18a] J. Sautter et al. "Evaluation of different excitation generation algorithms for artificial bandwidth extension". In: *2018 Elektronische Sprachsignalverarbeitung (ESSV)*. Ulm, Germany, Mar. 2018.

[Sau+18b] Jonas Sautter et al. "Discriminative training of deep regression networks for artificial bandwidth extension". In: *2018 IEEE International Workshop on Acoustic Signal Enhancement (IWAENC)*. IEEE, Sept. 2018.

[Sau+19] Jonas Sautter et al. "Artificial bandwidth extension using a conditional generative adversarial network with discriminative training". In: *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, May 2019, pp. 7005–7009.

[Sch33] K. O. Schmidt. "Neubildung von unterdrückten Sprachfrequenzen durch ein nichtlinear verzerrendes Glied". In: *Telegraphen- und Fernsprech-Technik* 22.1 (1933). (in German), pp. 13–22.

[SE18] Konstantin Schmidt and Bernd Edler. "Blind bandwidth extension based on convolutional and recurrent deep neural networks". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, Apr. 2018, pp. 5444–5448.

[SFS18] J. Sautter, F. Faubel, and G. Schmidt. "Feature selection for DNN-based bandwidth extension". In: *2018 Jahrestagung für Akustik (DAGA)*. Munich, Germany, 2018.

[SJV18] T. Schlien, P. Jax, and P. Vary. "Acoustic tube interpolation for spectral envelope estimation in artificial bandwidth extension". In: *Speech Communication; 13th ITG-Symposium*. 2018, pp. 1–5.

[Smi18] Leslie N. Smith. "A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay". In: *CoRR* (Mar. 26, 2018).

[Sri+14] N. Srivastava et al. "Dropout: a simple way to prevent neural networks from overfitting". In: *Journal of Machine Learning Research* 15 (2014), pp. 1929–1958.

[STS18] Yuki Saito, Shinnosuke Takamichi, and Hiroshi Saruwatari. "Statistical parametric speech synthesis incorporating generative adversarial networks". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 26.1 (Jan. 2018), pp. 84–96.

[SVN37] S. S. Stevens, J. Volkmann, and E. B. Newman. "A scale for the measurement of the psychological magnitude pitch". In: *The Journal of the Acoustical Society of America* 8.3 (Jan. 1937), pp. 185–190.

[TAL14] J. Tang, S. Alelyani, and H. Liu. "Feature selection for classification: A review". In: *Data Classification: Algorithms and Applications* (2014), p. 37.

[TIA06] TIA IS 127-B. *Enhanced variable rate codec, speech service options 3 and 68 for wideband spread spectrum digital systems*. Tech. rep. IS 127-B. TIA, 2006.

[TIA07] TIA IS 127-C. *Enhanced variable rate codec, speech service options 3, 68, and 70 for wideband spread spectrum digital systems*. Tech. rep. IS 127-C. TIA, 2007.

[TIA99] TIA IS 127. *Enhanced variable rate codec*. Tech. rep. IS 127. TIA, 1999.

[UCL18] UCL Division of psychology and language sciences. *PALS1004 Introduction to speech science - Week 6: Consonants*. 2018. URL: `https://www.phon.ucl.ac.uk/courses/spsci/iss/week6.php` (visited on 04/15/2020).

[Vin+10] Pascal Vincent et al. "Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion". In: *Journal of Machine Learning Research* 11 (Dec. 2010), pp. 3371–3408.

[VZY06] S. Vaseghi, E. Zavarehei, and Q. Yan. "Speech bandwidth extension: Extrapolations of spectral envelope and harmonicity quality of excitation". In: *2006 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*. Vol. 3. IEEE, June 2006.

[Wan+16] Yingxue Wang et al. "Speech bandwidth extension using recurrent temporal restricted boltzmann machines". In: *IEEE Signal Processing Letters* 23.12 (Dec. 2016), pp. 1877–1881.

[Wan+18] Ke Wang et al. "Investigating generative adversarial networks based speech dereverberation for robust speech recognition". In: *Proceedings of Interspeech, 2018, pp. 1581-1585* (Mar. 27, 2018).

[YA94] Yuki Yoshida and Masanobu Abe. “An algorithm to reconstruct wideband speech from narrowband speech based on codebook mapping”. English. In: *1994 International Conference on Spoken Language Processing (ICSLP)*. Sept. 1994, pp. 1591–1594.