UNIVERSITÀ DEGLI STUDI
DI TRENTO

DEPARTMENT OF INFORMATION ENGINEERING AND COMPUTER SCIENCE
ICT International Doctoral School

Deep Learning for Distant Speech Recognition

Mirco Ravanelli

Advisor:

Maurizio Omologo

Fondazione Bruno Kessler

December 2017

Acknowledgements

This achievement would not have been possible without the help and support of many people. Firstly, I would like to express a special appreciation to my advisor Maurizio Omologo, that has been an exceptional mentor for me. I would like to thank him for encouraging my research and for allowing me to grow as a research scientist. Besides my advisor, I would also like to thank all the members of the SHINE research unit for their scientific and technical support in the last years.

Special thanks to Prof. Yoshua Bengio, for hosting and advising me during my stay at the MILA lab. It was really a pleasure working with him and with the other top deep learning scientists of his amazing lab. In particular, I would like to mention Philemon Brakel, co-author of several papers I published during the last year, for his advice, support, and helpful discussions.

I would like to thank Prof. Laface and Prof. Squartini for reviewing this thesis, Prof. Sebe for being a committee member, as well as the FBK and UNITN administration staff for their availability and professionalism.

This PhD would not have been possible without my family. I would like to thank my wife Serena. Her encouragement, patience, and love were fundamental to face both beautiful and bad life moments. Last but not the least, I would like to express my sincere gratitude to my parents. They did multiple sacrifices to provide me a top-level education and their constant support over the years was priceless.

Abstract

Deep learning is an emerging technology that is considered one of the most promising directions for reaching higher levels of artificial intelligence. Among the other achievements, building computers that understand speech represents a crucial leap towards intelligent machines.

Despite the great efforts of the past decades, however, a natural and robust human-machine speech interaction still appears to be out of reach, especially when users interact with a distant microphone in noisy and reverberant environments. The latter disturbances severely hamper the intelligibility of a speech signal, making Distant Speech Recognition (DSR) one of the major open challenges in the field.

This thesis addresses the latter scenario and proposes some novel techniques, architectures, and algorithms to improve the robustness of distant-talking acoustic models. We first elaborate on methodologies for realistic data contamination, with a particular emphasis on DNN training with simulated data. We then investigate on approaches for better exploiting speech contexts, proposing some original methodologies for both feed-forward and recurrent neural networks. Lastly, inspired by the idea that cooperation across different DNNs could be the key for counteracting the harmful effects of noise and reverberation, we propose a novel deep learning paradigm called "network of deep neural networks".

The analysis of the original concepts were based on extensive experimental validations conducted on both real and simulated data, considering different corpora, microphone configurations, environments, noisy conditions, and ASR tasks.

Keywords

Deep Learning, Distant Speech Recognition, Deep Neural Networks.

Contents

Abbreviations

ACW Asymmetric Context Window

ADAM Adaptive Moment Estimation

AEC Acoustic Echo Cancellation

AI Artificial Intelligence

AM Acoustic Model

AMI Advanced metering infrastructure

AMIDA Augmented Multi-party Interaction with Distance Access

ANN Artificial Neural Network

ASA Auditory Scene Analysis

ASR Automatic Speech Recognition

BN Batch Normalization

CASA Computational Auditory Scene Analysis

CD Context Dependent

CE Cross-Entropy

CI Context Independent

CIFAR Canadian Institute for Advanced Research

CMVN Cepstral Mean and Variance Normalisation

CNN Convolutional Neural Network

CT Close-Talk

CTC Connectionist Temporal Classification

CW Context Window

DCASE Detection and Classification of Acoustic Scenes and Events

DICIT Distant-talking Interfaces for Control Interactive TV

DIRHA Distant-speech Interaction for Robust Home Applications

DNN Deep Neural Network

DRR Direct to Reverberant Ratio

DSR Distant Speech Recognition

ESS Exponential Sine Sweep

FA Frame Accuracy

FBANK Filter Bank

FBK Fondazione Bruno Kessler

FF-DNN Feed-Forward Deep Neural Network

fMLLR Feature space Maximum Likelihood Linear Regression

GAN Generative Adversarial Network

GCC-PHAT Generalized Cross Correlation with Phase Transform

GCF Global Coherence Field

GD Gradient Descend

GMM Gaussian Mixture Model

GRU Gated Recurrent Units

H-DNN Hierarchical Deep Neural Network

HLDA Heteroscedastic Linear Discriminative Analysis

HMM Hidden Markov Model

IM Image Method

IR Impulse Response

IWSLT International Workshop on Spoken Language Translation

LDA Linear Discriminative Analysis

Li-GRU Light Gated Recurrent Units

LIMABEAM Likelihood Maximization Beamforming

LJ Lehmann-Johansson algorithm

LM Language Model

LMS Least Mean Square

LPC Linear Predictive Coding

LSS Linear Sine Sweep

LSTM Long Short Term Memory

M-GRU Minimal Gated Recurrent Units

MAP Maximum a Posteriori

MBR minimum Bayes risk

MFCC Mel Frequency Cepstral Coefficients

MLP Maximum Likelihood Estimation

MLP Multi-Layer Perceptron

MLS Maximum Length Sequence

MMI maximum mutual information

MPE minimum phone error

MSE Mean Squared Error

NIST National Institute of Standards and Technology

NLL Negative Log-Likelihood

NN Neural Network

PCA Principal Component Analysis

PDT Phonetic Decision Tree

PER Phone Error Rate

PLP Perceptual linear prediction

PoV Probability of Voicing

PT Pre-Training

RBM Restricted Boltzmann Machines

ReLU Rectified Linear Units

ResNET Residual Neural Network

RNN Recurrent Neural Network

ROVER Recognition Output Voting Error Reduction

SBSS Semi-Blind Source Separation

SCW Symmetric Context Window

SGD Stochastic Gradient Descend

SNR Signal-to-Noise Ratio

SVM Support Vector Machines

Tanh Hyperbolic Tangent

TDNN Time Delay Neural Networks

TDOA Time Difference of Arrival

ToF Time of Flight

UBM Universal Background Model

VTLN Vocal Tract Length Normalization

WER Word Error Rate

WFST Weighted Finite State Transducers

WSJ Wall Street Journal

Publication List

This thesis is based on the following publications:

1. **M. Ravanelli**, P. Brakel, M. Omologo, Y. Bengio, "Light Gated Recurrent Units for Speech Recognition", in IEEE Transactions on Emerging Topics in Computational Intelligence (to appear).

2. **M. Ravanelli**, P. Brakel, M. Omologo, Y. Bengio, "A network of deep neural networks for distant speech recognition", in Proceedings of ICASSP 2017 (best IBM student paper award).

3. **M. Ravanelli**, P. Brakel, M. Omologo, Y. Bengio, "Improving Gated Recurrent Units by Revising Gated Recurrent Units", in Proceedings of Interspeech 2017.

4. **M. Ravanelli**, P. Brakel, M. Omologo, Y. Bengio, "Batch-normalized joint training for DNN-based distant speech recognition", in Proceedings of STL 2016.

5. **M. Ravanelli**, P. Svaizer, M. Omologo, "Realistic Multi-Microphone Data Simulation for Distant Speech Recognition", in Proceedings of Interspeech 2016.

6. M. Matassoni, **M.Ravanelli**, S. Jalalvand, A. Brutti, "The FBK system for the CHiME-4 challenge", in Proceedings of the CHiME 4 challenge.

7. **M. Ravanelli**, M. Omologo, "Contaminated speech training methods for robust DNN-HMM distant speech recognition", in Proceedings of INTERSPEECH 2015.

8. **M. Ravanelli**, L. Cristoforetti, R. Gretter, M. Pellin, A. Sosi, M. Omologo, "The DIRHA-English corpus and related tasks for distant-speech recognition in domestic environments", in Proceedings of ASRU 2015.

9. E. Zwyssig, **M. Ravanelli**, P. Svaizer, M. Omologo, "A multi-channel corpus for distant-speech interaction in presence of known Interferences", in Proceedings of ICASSP 2015.

10. **M. Ravanelli**, B. Elizalde, J. Bernd, G. Friedland, "Insights into Audio-Based Multimedia Event Classification with Neural Networks", in Proceedings of ACM-MMCOMMONS.

11. **M. Ravanelli**, M. Omologo, "On the selection of the impulse responses for distant-speech recognition based on contaminated speech training", in Proceedings of INTERSPEECH 2014.

12. L. Cristoforetti, **M. Ravanelli**, M. Omologo, A. Sosi, A. Abad, M. Hagmueller, P. Maragos, "The DIRHA simulated corpus", in Proceedings of LREC 2014.

13. M. Matassoni, R. Astudillo, A. Katsamanis, **M. Ravanelli**, "The DIRHA-GRID corpus: baseline and tools for multi-room distant speech recognition using distributed microphones", in Proceedings of INTERSPEECH 2014.

14. A. Brutti, **M. Ravanelli**, M. Omologo, "SASLODOM: Speech Activity detection and Speaker LOcalization in DOMestic environments", in Proceedings of Evalita 2014.

15. A. Brutti, **M. Ravanelli**, P. Svaizer, M. Omologo, "A speech event detection and localization task for multiroom environments", in Proceedings of HSCMA 2014.

16. **M. Ravanelli**, V.H. Do, A. Janin, "TANDEM-Bottleneck Feature Combination using Hierarchical Deep Neural Networks", in Proceedings of ISCSLP 2014.

17. **M. Ravanelli**, B. Elizalde, K. Ni, G. Friedland, "Audio Concept Classification with Hierarchical Deep Neural Networks", in Proceeding of EUSIPCO 2014.

18. B. Elizalde, **M. Ravanelli**, K. Ni, D. Borth, G. Friedland , "Audio-Concept Features and Hidden Markov Models for Multimedia Event Detection", in Proceedings of SLAM 2014.

Chapter 1

Introduction

The human voice is the most natural way to communicate. Building computers that understand speech thus represents a crucial step towards easy-to-use human-machine interfaces [201]. According to analysts, the global speech recognition market is estimated to be in order of $6 billions, with a tremendous growth expected in the next few years. As a matter of fact, voice-based interfaces are rapidly gaining a central role in our everyday lives. Current application areas include web-search, intelligent personal assistants, home automation, automotive, healthcare, financial transactions, consumer electronics, just to name a few. The growing interest in speech recognition is also witnessed by the remarkable investments of the most important tech companies, that have recently committed huge resources in the field. As a result, systems like Siri, Cortana, and Google Voice, gained considerable popularity and are currently used by millions of people worldwide.

The widespread diffusion of such commercial software, however, has given rise to the mistaken belief that speech recognition is a mostly solved problem. Actually, thanks to deep learning, modern speech recognizers achieve an unprecedented performance level. Nevertheless, despite the optimism and excitement surrounding ASR technologies, the road towards

a natural and flexible human-machine interaction is still long and full of scientific challenges. At the time of writing, some prominent open issues regard, for instance, the ability of dealing with very large vocabularies, better managing multiple languages with low resources, and properly modeling spontaneous speech. The development of real-time, small footprint and privacy-preserving systems also represents a crucial need. Another important problem concerns robustness against variability factors, including speaker, accent and channel variabilities.

This thesis considers the latter issue, addressing in particular the robustness against noise and reverberation, which we believe is one of the major open challenges in ASR. These disturbances typically arise when the speaker interacts with a distant microphone, making DSR a topic of fundamental interest for the research community. The research reported in this thesis is focused on deep learning, that is the natural candidate for improving DSR robustness, thanks to a more detailed modeling of the underlying properties of speech and acoustic environments.

The next section introduces deep learning, while the problem of distant speech recognition is summarized in Sec. 1.2. Sec. 1.3 then discusses the motivations and the scope of this research effort. Our contribution is summarized in Sec. 1.4, while Sec. 1.5 finally provides an outline of the thesis.

1.1 Deep Learning

Building intelligent machines has fascinated humanity for centuries [26]. The field of Artificial Intelligence (AI), however, started to develope relatively recently, when programmable digital computers were conceived. The history of AI has continuously alternated decades of great enthusiasm, investments and expectations, often followed by "AI winters", characterized

by reduced fundings and excitement towards this field [189]. The rise of deep learning [92] has recently contributed to renew the interest in AI and has allowed current technology to achieve higher levels of artificial intelligence. This paradigm has rapidly become a driving factor in academic and industrial research, and it is now being deployed in a wide range of domains, including computer vision (for object recognition, restoring colors in black and white photos, cancer detection, handwriting recognition, video classification), machine translation, as well as in natural language processing (for dialogue systems, question answering, image captioning, automatic writing) and speech recognition [297]. Other interesting applications are recommendation systems, fraud and risk detection, resource planning, predictions on financial markets, automatic game playing, robotics, self-driving cars, just to name a few.

Deep learning is actually a very general machine learning paradigm that follows a compositionality principle to represent the world around us efficiently. Current deep learning implementation exploits deep neural networks, that are properly trained to progressively discover complex representations starting from simpler ones. This principle can be applied in several practical problems, including the problem of recognizing distant speech, that will be briefly introduced in the following section.

1.2 Recognizing Distant Speech

Most of the current speech recognizers are based on a close-talking interaction with a microphone-equipped device, such a smartphone, a tablet, a laptop or even a smart watch. Although this approach usually leads to better performance, it is easy to predict that, in the future, users will prefer to relax the constraint of handling or wearing any device to access speech recognition services. Distant speech recognition could indeed represent the

preferred modality for future human-machine communications, especially in some specific contexts, where a distant interaction is more natural, convenient and attractive [290]. For instance, applications such as meeting transcriptions and smart TVs have been studied over the past decade in the context of the AMI/AMIDA [227] and the DICIT [191] projects, respectively. More recently, speech-based domestic control gained a lot of attention [150, 232]. To this end, the EU DIRHA project developed voice-enabled automated home environments based on distant-speech interaction in different languages [51, 216]. The recent success of commercial products like Amazon Echo and Google Home further confirms the great interest towards DSR in a domestic environment. Another possible application is distant speech interaction in operating theaters [224], in which the surgeon can dictate some notes about the operation or can access the medical records of the patient. Robotics, finally, represents another emerging application, where users can freely dialog with distant mobile platforms.

Under such hands-free scenarios, multiple issues arise that radically change the speech recognition problem. The distance between speaker and microphone tremendously degrades the intelligibility of the speech, hampering the performance of a speech recognizer. The recorded signal, in fact, not only accounts for the contribution of the desired speech, but also includes background noise, competing speakers, and non-speech acoustic events. Moreover, every acoustic enclosure introduces reverberation, that is originated by multiple delayed reflections on the room surfaces.

The research of the last decades led to an impressive improvement of these technologies [290]. However, a natural, robust and flexible speech interaction is far from being reached, especially in adverse environments [109]. For instance, even the most advanced state-of-the-art systems work relatively well in rather quiet environments, but often fail when facing more challenging acoustic conditions. This further indicates that several research

efforts are still required to continue the maturation of this technology.

1.3 Motivation and Scope

The main motivation behind this work is the belief that building machines able to naturally interact with humans is of paramount importance for reaching higher levels of artificial intelligence.

Recognizing speech is a very basic task for human beings. This apparent simplicity has contributed to create huge expectations around ASR, highlighting a significant gap between the possibilities offered by current technology and user requirements. A fundamental motivation is to contribute to bridge this gap, allowing future users to use speech technologies without current limitations and constraints.

The scope of this thesis is thus to explore proper techniques for improving the robustness of a DSR system to noise and reverberation with deep learning. More precisely, our goal is to properly revise standard deep learning principles, architectures and algorithms to better address DSR under such adverse conditions.

1.4 Contributions

DSR is a very active research field, that offers a huge literature on techniques, methodologies and algorithms. In the following, the main findings and contributions of the research activity undertaken in this PhD are summarized:

- **Methods for data contamination**. Data are playing a fundamental role in deep learning. Developing proper solutions for data contamination is thus of great interest for the research community. In the speech recognition field, however, it is still not clear what are the

best practices for deriving realistic contaminated data. In this thesis, our contribution is the definition of a methodology for deriving high-quality simulated data for ASR purposes. Several efforts have thus been devoted to the characterization of the reverberation effects of typical acoustic enclosures considering both measured and synthetic impulse responses. An extensive comparison between real and simulated data has shown the effectiveness of the proposed approach. The satisfactory level of realism obtained with our methodology, allowed us to generate some realistic simulated datasets and released them at international level. Proper techniques for exploiting these high-quality simulations for DNN training have then been proposed. Examples are the study of close-talking labels and close-talking pre-training techniques for distant speech recognition with contaminated data.

- **Better exploiting time contexts**. The importance of speech contexts is clear to the research community since several decades. However, past HMM-GMM systems were basically inadequate to manage large time contexts and one of the main reason behind the success of deep learning in speech recognition is the natural ability of DNN to properly manage long-term speech information. This feature is particularly helpful for distant speech recognition, since embedding more information into the system can counteract the remarkable uncertainty originated by noise and reverberation. A contribution of this thesis is the study of proper methodologies and architectures for better exploiting time contexts for DSR. Our studies involved both feed-forward and recurrent neural networks. For the former architectures, we proposed the use of an asymmetric context window to counteract acoustic reverberation. We also studied hierarchical neural networks for processing longer time contexts. Finally, we revise standard Gated Recurrent Units (GRUs) to improve the context management

for Recurrent Neural Networks. As a result, the proposed model called Light-GRU (Li-GRU) turned out to be helpful to both improve the ASR performance and to reduce the training time.

- **Cooperative networks of deep neural networks**: In this thesis we focused on DNN cooperation, that we believe is a factor of paramount importance for solving complex problems. More precisely, we believe that a proper cooperation and interaction across the different modules of a DSR system could be an interesting strategy to fight uncertainty. Based on this vision, we developed a novel paradigm called "*networks of deep neural* networks", where speech recognition, speech enhancement and acoustic scene analysis DNNs are jointly trained and progressively learn how to communicate each other to improve their performance. The proposed architecture is unfolded using a strategy similar to that used for RNNs and it is trained with a variation of the back-propagation algorithm called *back-propagation through network*. The experimental results highlighted the effectiveness of this approach.

1.5 Thesis Outline

This thesis is organized as follows.

The state-of-the art in deep learning is summarized in Chapter 2. This Chapter will describe the basic algorithms, the most popular architectures and will provide a brief overview on deep learning history, recent evolution, and future perspective.

Chapter 3 then proposes an overview on the main state-of-the art technologies involved in distant speech recognition. First, the main challenges of DSR are summarized. Then, the basic components of a DSR systems (i.e., speech recognition, speech enhancement and acoustic scene analysis)

will be described.

The innovative contributions of this work are summarized in the remaining three chapters: the methods for contaminated speech training are detailed in Chapter 4, our studies on time contexts are illustrated in Chapter 5, while the network of deep neural network paradigm is described in Chapter 6. Finally, some conclusions are drawn in Chapter 7.

The thesis also includes three appendices describing in detail the adopted corpora, the considered microphone configurations, and the different setups used in the experiments.

Chapter 2

Deep Learning

Current computers are able to efficiently solve problems that can be described by a sequence of well-defined formal and logic rules, but have many difficulties in tackling tasks that are hard to formalize as a list of basic operations (e.g., recognize speech or objects in images). A research challenge of AI is to develop machines able to address the latter category of problems.

Artificial intelligence can be approached in various ways. *Knowledge-based* solutions rely on hard-coding logical rules in a formal language. Another approach, called *machine learning*, aims to directly acquire knowledge from data, without (ideally) any human effort. The performance of machine learning systems strongly depends on the representation of the input data. For many years the research in the field has focused on proper methods for feature extraction, transformation and selection [21]. In standard pattern recognition approaches [23], these features were normally hand-designed and later exploited by classification algorithms.

Differently to past approaches, the idea of *representation learning* is to jointly discover not only the mapping from input feature to output, but also the features itself [16].

Deep learning [92] follows the philosophy of representation learning and aims to progressively discover complex representations starting from sim-

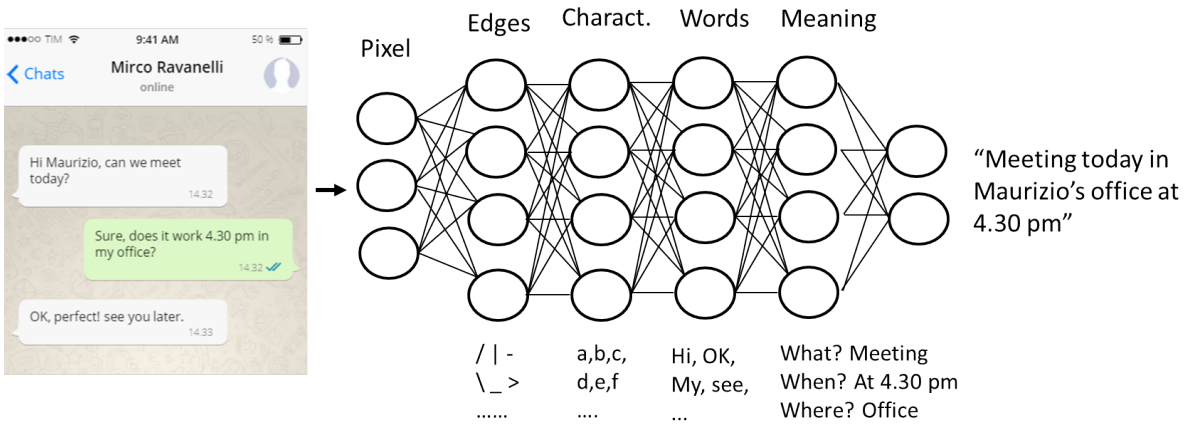


Figure 2.1: An example of an ideal deep learning system that learns hierarchical representations from low-level to higher level concepts.

pler ones. The principle of composition, on the other hand, can be used to describe the world around us efficiently. An example is reported in Figure 2.1, showing that combination of pixels can describe edges, from combination of edges we can derive characters, words, and, finally, the semantic meaning of a chat.

The deep learning paradigm is currently implemented with Deep Neural Networks (DNNs), that are Artificial Neural Networks (ANNs) based on several hidden layers between input and output. Each layer learns higher-level features that are later processed by the following layer [15]. When a suitable high-level representation is reached, a classifier can perform the final decision. Modern DNNs provide a very powerful framework for supervised learning: when adding more layers, in fact, a deep network can represent functions of increasing complexity and can potentially reach higher levels of semantic representations.

ANNs have been object of several research in the past decades [189]. These efforts were extremely important for the research community, since they laid the foundations for the basic learning algorithms [23]. For instance, the back-propagation algorithm was invented in the 60s-70s with

the contributions of many scientists [247]. Despite these achievements, the time was not yet ripe for the explosion of this technology. The current rise of deep learning can be explained with the following motivations:

- **Big Data**; A key ingredient for the success of this technology is the availability of large datasets. Current systems, in fact, aim to incorporate considerable knowledge into a machine, requiring lots of data. More precisely, the effectiveness of DNNs strongly depends on the capacity of the model, that can be increased by adopting deep and wide architectures. The improved network capacity, however, increases the number of parameters, inherently requiring more data to reliably estimate them. Fortunately, the rapid spread of internet and smartphones, allows easy and cheap big-data collections.

- **Computational power**; To properly exploit deep models and large datasets, a considerable computational power is required. In the last years, important progresses have been done to develop specialized hardware for deep learning. Modern Graphical Processor Units (GPUs), for instance, are currently used by most of deep learning practitioners to efficiently train complex models.

- **Computationally efficient inference**; DNNs usually require a lot of computational power during the training phase. An interesting aspect is that inference can be computed relatively efficiently, allowing, for instance, the development of real-time speech recognizers or low-latency dialogue systems.

- **Powerful priors**; Last but not least, deep learning incorporates reasonable assumptions about the world. The basic assumption is the compositionality principle previously discussed, that efficiently describes the complex world around us as a progressive composition

of different elements. This assumption acts as a prior knowledge used to defeat curse of dimensionality: among all the possible functions that are able to explain a dataset, deep learning restricts this selection to a smaller sub-set that satisfies the compositionality constraint. This naturally entails a regularization effect, that allows training deep architectures.

2.1 Basic Algorithms

This section proposes an overview of the main algorithms and techniques used for deep learning. In particular, some general notions about supervised learning are recalled in sub-section 2.1.1. The back-propagation algorithm is discussed in sub-section 2.1.2, while the main optimization techniques are summarized in sub-section 2.1.3. Regularization methods are finally described in sub-section 2.1.4.

2.1.1 Supervised Learning

DNNs are often trained in a supervised fashion [23]. This training modality can be formalized as follows: let's assume to have a set of N training examples $\{(x_1,y_1),...,(x_i,y_i),...,(x_N,y_N)\}$, where each element is a pair composed of a feature vector x_i and its corresponding label y_i. The learning algorithm seeks a function $f : X \rightarrow Y$ that maps the input space X into the output space Y. Deep learning is a form of parametric machine learning, whose function f depends on a set of trainable parameters θ. For each particular choice of θ, a different mapping function f is obtained. The number of possible functions that can be represented by the DNN is called *capacity*.

The goal of supervised learning is to find a function f that is able to "explain" well the training samples. More formally, this implies finding

proper values of θ able to minimize a certain performance metric:

$$\hat{\theta} = \arg\min_{\theta} L(Y, f(X, \theta)) \tag{2.1}$$

The function L is called *loss* (or cost) and, intuitively, should assume low values when the parameters θ lead to an output well-matching with the reference labels.

In the context of the Maximum Likelihood Estimation (MLE), the optimization problem can be reformulated in this way:

$$\hat{\theta} = \arg\max_{\theta} P(Y|X, \theta) \tag{2.2}$$

where $P(Y|X, \theta)$ is the conditional probability distribution defined at the output of the DNN. To perform a MLE estimation, a popular choice of L is the Negative Log-Likelihood (NLL) or Cross-Entropy (CE). In this case, the MLE optimization can be rewritten as:

$$\hat{\theta} = \arg\min_{\theta} -log(P(Y|X, \theta)) \tag{2.3}$$

Another popular choice is the Mean Squared Error (MSE):

$$\hat{\theta} = \arg\min_{\theta} ||Y - f(X, \theta)||^2 \tag{2.4}$$

Solving the training optimization problem is challenging, especially for DNNs composed of a huge number of parameters. The functions f originated by DNNs are, in fact, typically non-linear, making the optimization space highly non-convex. Recently, some theoretical and experimental studies have shown that the main challenge are *saddle points* and not local minima as commonly believed in the past [56]. Even though these results are still object of an open debate in the research community, they suggest that the parameter space is flat almost everywhere and the (very rare) local minima are almost all also global ones [134].

There are in principle various ways to solve this optimization problem [28]. A naive solution would be to try all the possible θ and choose the configuration that minimizes the cost function. This approach is clearly unfeasible, especially for a large number of parameters. Another solution would be to exploit evolutionary optimizations, based on genetic algorithms or particle swarm techniques [63]. Despite the interesting aspects of these optimizers (e.g, high parallelism, differentiability is not strictly required), such methods require very frequent evaluations of the cost function, that is impractical for DNNs trained on large datasets. At the time of writing, the most popular choice is gradient-based optimization. The computation of the gradient is described in the following section, while the optimization step will be discussed in Sec. 2.1.3.

2.1.2 Back-Propagation Algorithm

The gradient $\partial L/\partial\theta$ is a very precious information that describes what happens to the cost function L when a little perturbation is applied to the parameters θ. If this perturbation causes an improvement of the loss, it could be convenient to do a little step in the direction indicated by the gradient.

The computation of $\partial L/\partial\theta$ is normally performed with the back-propagation algorithm [237], that is often misunderstood as meaning the whole learning procedure. Actually, back-propagation is only a method for computing the gradient.

Deriving an analytical expression of $\partial L/\partial\theta$ is rather straightforward for systems that are differentiable almost everywhere. In most of the cases, in fact, the analytical expression of the gradient is a direct application of basic calculus rules [156]. As shown in Figure 2.2, a DNN can be described as a composite function that performs a chain of computations. Gradients can thus be computed with the chain rule [156], as reported in the following

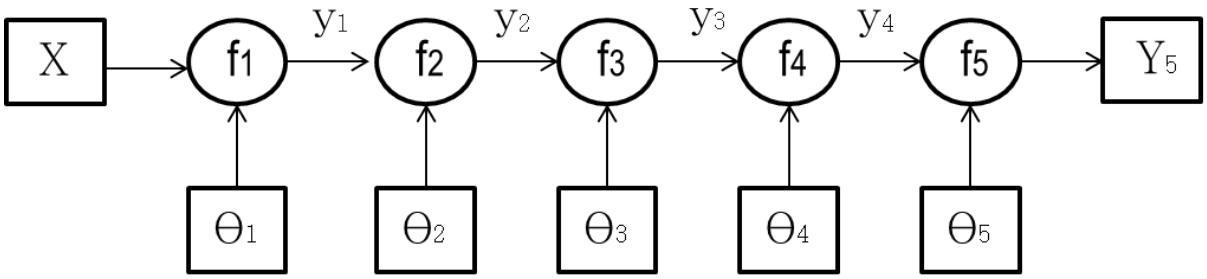


Figure 2.2: A composite function representing the computations performed in the various layers of a DNN.

equations:

$$\frac{\partial y_5}{\partial \theta_1} = \frac{\partial y_1}{\partial \theta_1} \cdot \frac{\boldsymbol{\partial y_2}}{\boldsymbol{\partial y_1}} \cdot \frac{\boldsymbol{\partial y_3}}{\boldsymbol{\partial y_2}} \cdot \frac{\boldsymbol{\partial y_4}}{\boldsymbol{\partial y_3}} \cdot \frac{\boldsymbol{\partial y_5}}{\boldsymbol{\partial y_4}} \tag{2.5a}$$

$$\frac{\partial y_5}{\partial \theta_2} = \frac{\partial y_2}{\partial \theta_2} \cdot \frac{\boldsymbol{\partial y_3}}{\boldsymbol{\partial y_2}} \cdot \frac{\boldsymbol{\partial y_4}}{\boldsymbol{\partial y_3}} \cdot \frac{\boldsymbol{\partial y_5}}{\boldsymbol{\partial y_4}} \tag{2.5b}$$

$$\frac{\partial y_5}{\partial \theta_3} = \frac{\partial y_3}{\partial \theta_3} \cdot \frac{\boldsymbol{\partial y_4}}{\boldsymbol{\partial y_3}} \cdot \frac{\boldsymbol{\partial y_5}}{\boldsymbol{\partial y_4}} \tag{2.5c}$$

$$\frac{\partial y_5}{\partial \theta_4} = \frac{\partial y_4}{\partial \theta_4} \cdot \frac{\boldsymbol{\partial y_5}}{\boldsymbol{\partial y_4}} \tag{2.5d}$$

Beyond the mathematical expression, the intuition behind the chain rule is this: the little change applied to the parameter θ_1 causes a change on the output of function f_1. This perturbation will propagate until the end of the chain, eventually causing a perturbation on the final node, that typically computes the cost function L.

Despite the relative simplicity of the gradient expression, its numerical evaluation can be computationally expensive. A naive (but inefficient) solution would be to independently evaluate the gradient expression for each parameter. The back-propagation algorithm [237] is a simple and inexpensive procedure that performs the various operations in a specific order. From the previous equations, one can notice that there are several shared computations. It would be, for instance, very convenient to start

from the last equation, store the results in bold, and reuse them to compute the following gradients.

The back-propagation algorithm is based on a dynamic programming approach and is summarized by the following step:

1. **Forward Propagation**; propagate the input features to the output and store the activations y.
2. **Compute Cost**; evaluate the cost function L at end of the chain.
3. **Back Propagation**; compute the gradient from the last element of the chain to the first one.

When propagating gradients through long computational chains, however, some issues might arise [197]. The chain rule, in fact, implies several gradient multiplications that can lead to *vanishing* or, less often, to *exploding gradients* [19].

Exploding gradients can effectively be tackled with simple clipping strategies [197], while vanishing gradient is more critical and might impair training very deep neural networks. As will be discussed in the following part of this chapter, special architectures are often needed to properly address this issue.

2.1.3 Optimization

Once computed, the gradient has to be exploited by an optimizer to progressively derive better parameters.

The most popular optimization algorithm is Gradient Descend (GD), that updates θ according to the following equation:

$$\theta = \theta - \eta \frac{\partial L}{\partial \theta} \tag{2.6}$$

The parameters are updated in the direction pointed by the gradient, with a step size determined by the learning rate η (the minus is due to the minimization of the loss). Note that gradient-based optimization does not provide any guarantee on global optimality. In such complex high-dimensional spaces, a crucial role is thus played by a proper initialization of the θ and by a suitable choice of loss and activation functions.

Depending on how many data are used to compute the gradient, some variants of GD can be defined. When the gradient is computed on the full training dataset, the optimizer is called batch-GD. A popular alternative is Stochastic Gradient Descend (SGD) that splits the dataset into several smaller chunks (called mini-batches) and update the parameters more frequently, with well-known benefits in terms of both accuracy and training convergence.

Standard SGD, however, has trouble navigating on areas where the surface curves much more steeply in one dimension rather than in another. To mitigate this issue, a momentum is often applied to the update equations for accelerating the training convergence [261].

Another issue is that the same learning rate applies to all the parameters. This stiffness can be critical, since each θ has its own characteristics, possibly requiring independent learning rates. For instance, if the absolute value of the gradient is very large in a specific dimension, it means that we are in a very steep area and would be more prudent to do a little step using a small learning rate. On the contrary, when the gradient is small, a higher learning rate can be used. Following this philosophy, several variations of SGD, such as Adagrad [61], Adadelta [299], RMSprop, Adam [137], have been recently proposed (see the reference papers for more details). These solutions, in general, rescale η by gradient-history metrics, resulting in a faster and more robust optimization.

2.1.4 Regularization

A crucial challenge in machine learning is the ability to perform well on previously unobserved inputs. This ability is called *generalization* [92]. Possible causes of poor generalization are *underfitting* and *overfitting*, that are two central issues when training DNNs. Underfitting occurs when the model is not able to reach a sufficiently low error on the training set. This might arise when the DNN has not enough capacity. Overfitting occurs when the gap between training and test errors is too large. Differently to undefitting, this might happen when the model has excessive capacity. The strategies to counteract overfitting are known as *regularization*, and normally consist of methods for reducing the network capacity based on prior knowledge. From this point of view, even deep learning itself can be regarded as a regularization technique, due to the prior knowledge naturally embedded in the compositionality principle.

The great importance of these methodologies has made regularization one of the major research directions in the field. In the following, the most popular regularizers are described.

L^2 Regularization

Many regularization approaches are based on limiting the model capacity by adding a parameter norm penalty to the cost function. These approaches tend to penalize too complex solutions, following the philosophy of the *Occam's razor* principle: "among competing hypotheses leading to the same performance, the simpler one should be selected". The most common method following this approach is the L^2 regularization, that add a penalty to the loss function L:

$$\tilde{L} = L + \alpha||\theta||^2 \tag{2.7}$$

where α is an hyperparameter that weights the contribution of the regularization term. The insight behind this kind of regularization is that solutions characterized by higher norms of the the trainable parameters are more complex and are more likely to overfit the training dataset.

Dropout

An effective way to improve generalization is to combine several different models [31]. If each classifier has been trained separately, it might have learned different aspects of the data distribution and their mistakes are likely to be complementary. Combining them helps produce a stronger model, that is hopefully less prone to overfitting. Even though these methods are very effective, a major limitation lies in the considerable computational efforts needed to train and test different DNNs. Dropout [257] is an effective regularization method that provides an inexpensive approximation to training and evaluating an exponential number of neural networks.

The key idea is to randomly drop neurons during training with a probability called dropout rate ρ. This way, a subnetwork is sampled from an exponential number of smaller DNNs for each training sample. At test time, the whole network is used (i.e., the DNN with all the neurons active), but the activations are scaled down by ρ.

This ensemble learning approach significantly reduces overfitting and gives major performance improvements. Many variants of dropout have been proposed in the literature [281, 39]. For instance, in [280] the regularizer is applied to the weight connections rather than on neurons, while in [182, 79] dropout is extended to recurrent neural networks.

Data Augmentation

The best way to achieve generalization would be to train the model with more data. However, in practice, the amount of data is limited and the

collection of large annotated corpora is very expensive. A possible alternative is to artificially process the available data, in order to generate novel training samples. For a computer vision application, one can for instance, rescale, rotate, or shift the available images to generate novel samples. For speech recognition, one way is to apply proper algorithms able to modify pitch, formants, and other speaker characteristics. Data augmentation can also be regarded as a way for adding prior knowledge to a model, since we exploit the information that the new samples do not change the class label when are processed.

Other Regularizers

Other approaches have been proposed for counteracting overfitting. A popular way is to add random noise during learning. Several methods have been proposed in the literature, proposing to add it at gradient, weight, input, output or hidden activation levels [106, 130, 169].

An alternative consists in adopting a semi-supervised approach, where unsupervised data can be exploited as prior knowledge to improve generalization. From this point of view, the pre-training approach based on Restricted Boltzman Machines (RBM) [117] or autoencoders [275] can be regarded as a form of regularization. Multi-task learning [40] (i.e., building DNN solving multiple correlated tasks) can also be considered as a sort of regularization, since it encourages the DNN to discover very general features at the first hidden layers.

Finally, one the most popular and simple approach for regularization is early stopping [22]. The idea is to periodically monitoring the performance of the DNN on held-out data and stop the training algorithm when this performance begins to deteriorate.

2.1.5 Hyperparameter selection

Most deep learning algorithms are based on some hyperparameters that must be properly set to ensure a good performance. The most important ones describe the structure of the network (e.g., number of hidden layers, number of hidden units per layer), determine the optimization characteristics (e.g., learning rate), and specify the behaviour of the regularizer (e.g., weight decay or dropout rate). Properly setting the hyperpameters is rather difficult, mainly because a new model should be trained for each new setting. Moreover, several hyperparameters can be correlated each other, making an independent optimization of them usually not viable.

There are two basic approaches to derive them: choosing them manually and choosing them automatically. The manual selection requires a considerable experience and familiarity with the addressed task as well as a precise knowledge of the role of each specific hyperparameter. This is possible for well-explored machine learning tasks, where a detailed literature suggesting reasonable settings is available.

When the manual approach is not feasible, a possible alternative is to automatically select the hyperparameters with grid search. Grid search simply tries all the combinations over a specified range of values. Although this search can be easily parallelized, its computational expense is exponential in the number of hyper-parameters. A straightforward alternative is to sample them randomly [20]. This approach is still very easy to parallelize, and is generally faster than grid search.

2.2 Main Architectures

This section summarizes the main architectures used in deep learning. In particular, sub-sec. 2.2.1 discusses the main type of neurons. Sub-sec. 2.2.2 describes feed-forward neural networks, sub-sec. 2.2.3 introduces batch

normalization, while sub-sec. 2.2.4 describes Recurrent Neural Networks (RNNs). Finally, some architectural variations recently proposed in the literature are summarized in sub-sec. 2.2.5.

2.2.1 Neuron Activations

The computations performed by the DNNs are a sequence of linear operations followed by non-linear ones. More precisely, the output of a hidden layer h_{i+1} composed of n neurons and fed by m input features h_i, can be represented as follows:

$$h_{i+1} = g(\underbrace{Wh_i + b}_{a}) \tag{2.8}$$

where W is the weight matrix $(n \times m)$, b is the bias vector of n element and g is the activation function. The linear transformation performed before applying g is called affine transformation a. The choice of g is particularly important, and several research efforts have been devoted to study proper activation functions. The most popular are:

- **Linear**; The simplest neurons are obtained by directly taking the affine transformation without any non-linearity. These neurons are called linear and can be used, for instance, for predicting real numbers in the output layer (regression problem).

- **Sigmoid**; For several decades, the most popular activation was the logistic sigmoid:

$$\sigma(x) = \frac{1}{1 + exp(-x)} = \frac{exp(x)}{1 + exp(x)} \tag{2.9}$$

  The main advantage is that such activations are bounded between 0 and 1. However, the use of sigmoid activations in modern feedforward networks is now discouraged, since they are characterized by close-to-zero gradients across most of their domain. In fact, $\sigma(x)$ saturates

when their input is very positive or very negative, slowing down the training.

- **Hyperbolic Tangent**; This kind of non-linearity is a rescaled version of the sigmoid function:

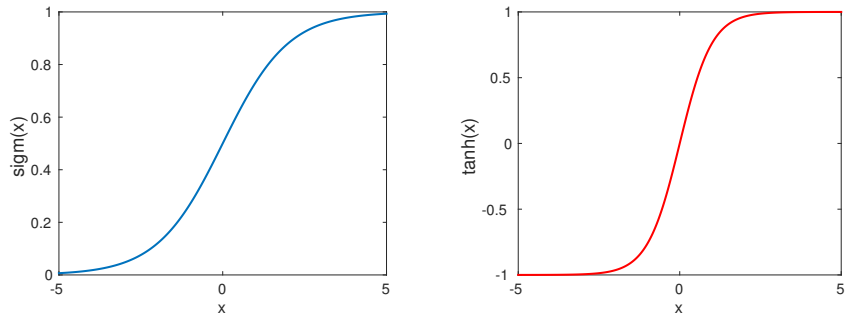
$$tanh(x) = \frac{exp(x) - exp(-x)}{exp(x) + exp(-x)} = 2\sigma(2x) - 1 \tag{2.10}$$

It suffers from the main disadvantages of sigmoid units, but in general it provides better performance. This is due to the fact that tanh is symmetric around zero, making this activation less prone to saturation in the last layers.

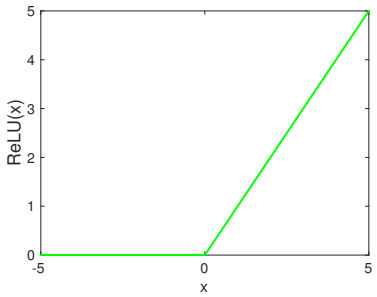
- **Rectified Linear Units**; This kind of activation is currently the most popular choice in modern feed-forward neural networks [127, 300]:

$$ReLU(x) = max(0, x) \tag{2.11}$$

The main insight behind the use of these activations is that they are rather similar to linear units. Linear units, as we have outlined before, lead to a convex optimization space, that is very easy to optimize. ReLUs inherit, at least in part, the benefits of the latter activations. More precisely, the derivatives of a ReLU remain large whenever the unit is active. When initializing the parameters of the affine transformation, a good practice is to set the bias b to a small positive value (e.g., 0.1), fostering the unit to be active during the first part of the training. This allows the derivatives to pass unaltered through all the non-linearities without vanishing gradient problems. The main issue is that the unboundedness of ReLU can cause large activations with possible problems of numerical stability. Several modifications to standard ReLU have been proposed in the literature, including leaky ReLU [164], parametric ReLU [111] and Maxout units [94].



(a) Sigmoid (b) Tanh



(c) ReLU

Figure 2.3: Main activation functions used in deep neural networks.

- **Softmax**; Softmax neurons are often used in the output layer to estimate a probability distribution over a set of n alternatives. To represent probabilities, each output neuron must assume values between 0 and 1, and the sum of all of them must be 1. Formally, the softmax of the i-th neuron is defined as follows:

$$softmax(a_i) = \frac{exp(a_i)}{\sum_{j=1}^{n} exp(a_j)} \tag{2.12}$$

Softmax creates a competition across the neurons through its normalization of the affine transformation a. This normalization can be done, in principle, with many other functions, including the linear one. The choice of the exponential function is done because it couples well with the maximum log likelihood optimization carried out to train DNNs.

The log-softmax, in fact, can be written as follows:

$$log\big(softmax(a_i)\big) = a_i - log\big(\sum_{j=1}^{n} exp(a_j)\big) \tag{2.13}$$

The MLE optimization would try to push up the activation of correct neurons, while penalizing the other ones, in particular the most active incorrect predictions.

When training DNNs, a crucial aspect is the parameter initialization, that has to be carefully designed according to the specific choice of the activation function. A crucial requirement is to break the symmetry (i.e., avoiding setting all the parameters with the same initial value). In fact, same weights will compute the same gradients with identical updates. A popular approach is to randomly initialize the parameters with small random real values. One problem with this method is that the variance at the output of each neuron grows with the number of input weights. As proposed in [89], one solution is to normalize the initial weight variance by the number of neurons.

2.2.2 Feed-forward Neural Networks

According to their architecture, DNNs can be classified into two main categories: feed-forward and recurrent neural networks. In the context of Feed-Forward DNNs (FF-DNN), the prediction $P(y/x)$ is just a function of the current input x and of the parameters θ:

$$y = f(x, \theta) \tag{2.14}$$

These models are called feed-forward because the information flows from the input to the output without any feedback connection. An example of feed-forward neural network composed of four hidden layers is shown in Figure 2.4. The figure depicts a popular architecture called fully-connected

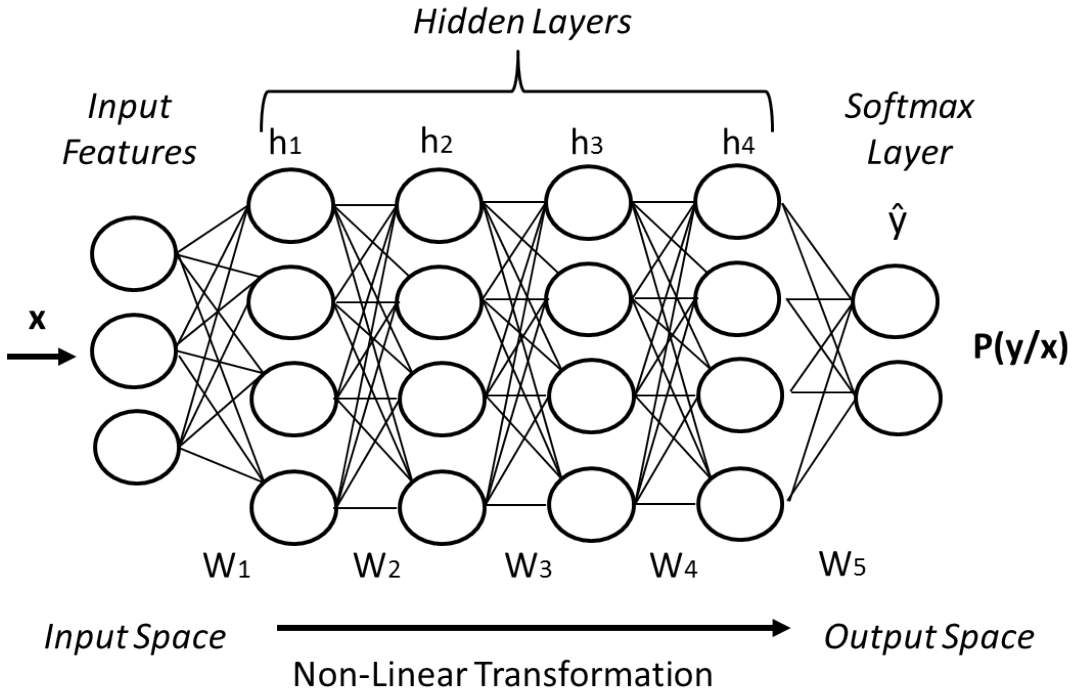


Figure 2.4: A Multi-Layer Perceptron (MLP) composed of four hidden layers.

DNN or Multi-Layer Perceptron (MLP), where all the neurons are connected with the ones of the following layer.

Figure 2.5 shows an alternative representation of the network, using the so-called computational graphs, that are flexible tools for formalizing sequences of computations. There are several ways of formalizing computational graphs. In the figure, for instance, the variables (that can be scalars, matrices or tensors) are represented by squares, while the operations are represented by circles. This approach can be extended to describe all the DNN computations, including the operations performed for gradient and cost function estimation.

Interesting, the universal approximation theorem states that a MLP with at least one hidden layer can approximate any continuous function[1]

[1]The theorem is actually restricted to continuous functions on compact sets that operates a mapping from a finite-dimensional space to another.

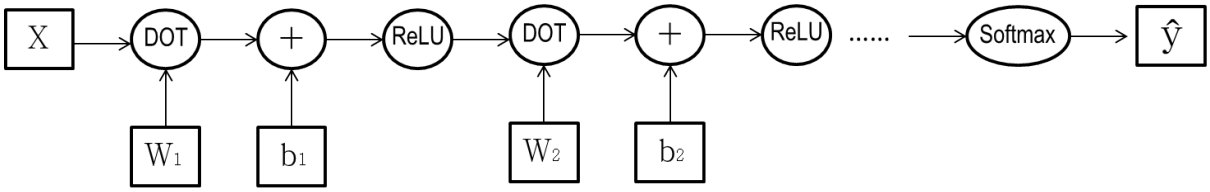


Figure 2.5: A computational graph representing the chain of computation for a feed-forward MLP.

with any desired non-zero error, provided that the network is composed of enough hidden neurons [120]. This theorem seems in contrast with the deep learning principle, which suggests to adopt DNNs composed of several hidden layers. However, the universal approximation theorem only guarantees that the function exists, but does not provide any insight on how to find it. There are thus no guarantees that the training algorithm will be able to learn that function. Moreover, the theorem does not provide any way for properly choosing this transformation such that it generalizes to points not included in the training set. On the other hand, another theorem (called "*no free lunch* theorem [291]) says that there is no universally superior machine learning algorithm.

An alternative to fully-connected neural neural networks are Convolutional Neural Networks (CNNs) [151]. Differently to the former ones, CNNs are based on local connectivity, weight sharing and max pooling. The combination of these characteristics make CNNs particularly suitable for managing correlations across features. Moreover, the presence of max pooling allows the network to obtain shift-invariant properties. CNNs are inspired by biological studies of the visual cortex and are extremely successful in practical applications, especially in computer vision [145]. CNNs have also been applied in speech recognition, achieving interesting performance thanks to some degree of invariance to small shifts of speech features along the frequency axis, that resulted important to deal with speaker and

environment variations [1].

2.2.3 Batch normalization

Training DNNs is complicated by the fact that the distribution of each layer's inputs changes during training, as the parameters of the previous layers change. This problem, known as *internal covariate shift*, slows down the training of deep neural networks. Batch normalization [123], that has been recently proposed in the deep learning community, addresses this issue by normalizing the mean and the variance of each layer's pre-activation for each training mini-batch. It has been long known that the network training converges faster if its inputs are properly normalized [152] and, in such a way, batch normalization extends this normalization to all the layers of the architecture. More precisely, batch normalization is defined as follows:

$$BN(a) = \gamma \frac{a - \mu_b}{\sqrt{\sigma_b^2 + \epsilon}} + \beta \tag{2.15}$$

where a is the neuron pre-activation (i.e., the output before applying the non-linearity), μ_b and σ_b are the mean and standard deviations computed for each minibatch and ϵ is constant introduced for numerical stability. The variables γ and β are trainable scaling and shifting parameters, introduced to allow each neuron of the network to have an explicit control of mean and variance statistics. The computations involved for batch normalization are fully differentiable, and it is possible to back-propagate through it.

Batch normalization resulted particularly helpful to achieve regularization, to significantly speed-up the convergence of the training phase as well as to improve the overall accuracy of a DNN. The regularization effect is due to the fact that mean and variance normalizations are performed on each mini-batch rather than on the entire dataset. This approximated estimation of the normalization statistics introduces a sort of noise in the

learning process, that resulted helpful to counteract overfitting [148, 49]. Moreover, batch normalization inherently reduces the capacity of the DNN: among all the possible functions able to explain the training data, only the subset that respect the normalization constraint can be chosen.

2.2.4 Recurrent Neural Networks

Recurrent Neural Networks (RNN) are architectures suitable for processing sequences [158]. The elements of a sequence are, in most of the cases, not independent. This means that, in general, the emission of a particular output might depend on the surrounding elements or even on the full-history. To properly model the sequence evolution, the presence of memory to keep track of past or future elements is thus of fundamental importance. The memory can be implemented using feedback connections, that introduce the concept of *state*. In particular, RNNs are characterized by the following equation:

$$h_t = f(x_t, h_{t-1}, \theta) \tag{2.16}$$

where h_t and h_{t-1} are the current and the previous states, respectively. Due to this recurrence, the current state actually depends on all the previous ones.

The simplest form of RNN is the so-called *vanilla* RNN, that is described by the following equation:

$$h_t = tanh(Wx_t + Uh_{t-1} + b) \tag{2.17a}$$

$$\tag{2.17b}$$

In this case, the parameters θ are the weight matrix W (feed-forward connections), the matrix U (recurrent weights) and the vector b (bias).

To train RNNs, an unfolding of the network is necessary. This operation, that is graphically represented in Figure 2.6, can be carried out as

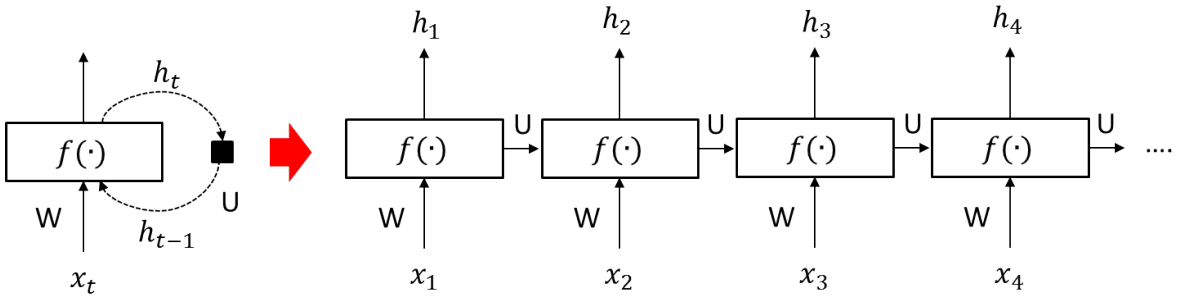


Figure 2.6: An example of RNN unfolding procedure.

highlighted in the following equation:

$$h_3 = f(x_3, h_2, \theta) \tag{2.18a}$$

$$h_3 = f(x_3, f(x_2, h_1, \theta), \theta) \tag{2.18b}$$

$$h_3 = f(x_3, f(x_2, f(x_1, h_0, \theta), \theta), \theta) \tag{2.18c}$$

After unfolding, the RNN can be treated as feed-forward neural network, that is very deep along the time axes. Therefore, the same algorithms adopted for training feed-forward neural networks can be used. Sometimes, the back-propagation algorithm in the context of recurrent neural networks is called back-propagation through time, to emphasize that the gradient is propagated through the time axes. An important aspect of RNNs is that the parameters are shared across the time steps, making model generalization more easy.

Back-propagating the gradient through many time steps, however, can be complicated by vanishing and exploding gradients, that might impair learning long-term dependencies [19]. As we have seen in Sec. 2.1.2, exploding gradients can effectively be tackled with simple clipping strategies [197], while vanishing gradient requires special architectures to be properly addressed. A common approach relies on the so-called gated RNNs, whose core idea is to introduce a gating mechanism for better controlling the flow of the information through the various time-steps. Within this family of

architectures, vanishing gradient issues are mitigated by creating effective "shortcuts", in which the gradients can bypass multiple temporal steps.

The most popular gated RNNs are LSTMs [118], that often achieve state-of-the-art performance in several machine learning tasks, including speech recognition [97, 244, 4, 284, 45, 172]. LSTMs rely on a network design consisting of memory cells that are controlled by forget, input, and output gates (see [118] for the full list of equations describing this model). Despite their effectiveness, such a sophisticated gating mechanism might result in an overly complex model. On the other hand, computational efficiency is a crucial issue for RNNs and considerable research efforts have recently been devoted to the development of alternative architectures [100, 131, 302]. A noteworthy attempt to simplify LSTMs has recently led to a novel model called Gated Recurrent Unit (GRU) [46, 47], that is based on just two multiplicative gates. This typology of RNN has been object of several studies in this thesis, as will be discussed in sec. 5.3.

2.2.5 Architectural Variations

During the last years, several novel architectures have been proposed in the deep learning field. Although a complete overview of all the recently-proposed architecture is out of the scope of this thesis, this subsection summarizes the most popular ones.

First of all, architectures based on combinations of convolutional, recurrent and fully-connected layers gained a lot of popularity, especially in the field of speech recognition [243]. For ASR, convolutional layers are used for feature extraction, recurrent layers for temporal processing, and fully-connected layers for the final classification.

Other recently-proposed DNNs extend the shortcut idea to feed-forward DNN for improving the propagation of the gradient across the various hidden layers. With this regard, a popular architecture is called Residual Neu-

ral Networks (ResNet) [112]. ResNets consider a direct connection through the various hidden layers, that allow the gradient to flow unchanged. By stacking these layers, the gradient can theoretically pass over all the intermediate layers and reach the bottom one without being diminished. These networks are called residual because, instead of directly modeling the output distribution, they actually model the difference between the input and output distributions. When several hidden layers are available, the network representations evolve rather slowly from a low-level to a high-level features. It would be thus natural to assume that input and output distributions would be rather similar and could be convenient to model this small difference.

A related idea is exploited in the context of Highway connections [258, 302]. Highway Networks preserve the shortcuts introduced in ResNets, but augments them with a learnable gate that manages the information flow through the hidden layers. The latter networks actually extend the idea of multiplicative learning gates to feed-forward DNNs.

2.3 Deep Learning Evolution

In the following subsections a brief history, the recent progress as well as the future challenges of deep learning are summarized.

2.3.1 Brief History

To better understand deep learning, it is useful to briefly summarize its history. Contrary to common belief, deep learning is not a new research field and has a rather long and troubled history, starting approximatively in the 40s. During the past 80 years, deep learning changes its name (and fortune) several times. There were, indeed, three main historical trends that have characterized the evolution of this technology under different

names: cybernetics (40s-60s), connectionism (80s-90s) and deep learning (2006-).

During 40s, cybernetics was inspired by theories of biological learning. For instance, McCulloch-Pitts [170] proposed a first simplified model of a neuron. This neuron was able to recognize two different categories by testing whether the output (derived from a simple linear transformation) was positive or negative. Interestingly, this neuron resembles the ones used nowadays in modern deep learning systems, where the linear transformation processes the inputs of each hidden layer. In the model proposed by McCulloch-Pitts, the weights W were set by a human operator. The work by Rosenblatt [235], who led to a neuron called Perceptron, became the first model that could learn the weights from data in a supervised way. The adaptive linear element (ADALINE) proposed almost simultaneously by Widrow and Hoff [286] was able to exploit a linear neuron to predict a real number (regression problem), using an algorithm very similar to Stochastic Gradient Descend (SGD).

The second evolution of deep learning started with the connectionism approach (80s-90s) [23]. Connectionism was also strongly influenced by the studies on biological neurons and it is based on the idea that complex non-linear functions can be obtained from the combination of several simple artificial neurons. In effect, biological neurons are cells performing a basic operation and networking them is the key for archiving intelligence. A major accomplishment of the connectionist era was the popularization of the back-propagation algorithm [238]. During the 90s, important advances were made in modeling time sequences with recurrent neural networks. In particular, Hochreiter and Bengio [19] identified some of the fundamental challenges in modeling long sequences. Some years later, Hochreiter and Schmidhuber (1997) proposed a novel approach for mitigating some of these issues, leading to an architecture called Long Short Term Memory (LSTM)

[118] that is still very popular to process sequences. The interest towards connectionism began to wane since mid 90s, in favor of kernel methods [255] and graphical models [144], that achieved good performance in many tasks. During this long deep learning winter, the progress in the field was mainly promoted by the Canadian Institute for Advanced Research (CIFAR), that gathers the research labs led by Geoffrey Hinton the at University of Toronto, Yoshua Bengio at University of Montreal, and Yann LeCun at New York University.

The new life of neural networks began in 2006, when Geoffrey Hinton showed that neural networks with many hidden layers (called Deep Neural Networks) could be trained using an unsupervised pre-training strategy based on deep belief networks [117]. This milestone contributed to rekindle interest in deep learning, paving the way for the recent progress in the field that will be discussed in the following sub-section.

2.3.2 Recent Progress

The initial deep learning breakthrough of 2006, seemed to be suggesting that the only possible approach for training deep neural networks was to initialize them in an unsupervised way. Actually, shortly after Hinton's paper, it was shown that deep autoencoders were equally able to properly pre-training DNNs [275]. In the following years, it was gradually more clear that pre-training step was not strictly necessary to train deep models. In fact, several works showed that simply using ReLU [185] activations (more effective to fight vanishing gradients) with a proper weight initialization [89], allows one to avoid any pre-training step.

Moreover, noteworthy progresses have then been done in sequence-to-sequence learning. Before 2014, it was widely believed that this kind of learning would require labeling of each individual element of the sequence. In 2014, the study of attention mechanisms [9] has offered a flexible and

powerful framework able to overcome previous limitations, revolutionizing many fields, including machine translation. The same year, memory-augmented deep neural networks [99] have been successfully trained for learning simple programs from examples of desired behaviors (e.g., sort lists).

Important achievements have also been obtained in the context of reinforcement learning [262], where autonomous agents learn to solve tasks with a trial and error strategy. DeepMind showed that a deep reinforcement learning-based agent can learn to play Atari games, reaching human or super-human performance [179]. DeepMind also stunned the world when the AlphaGo system was able to defeat the world champion of Go, an ancient Chinese game with a huge number of possible moves (enormously larger than the combinations of chess) [67].

A key element for the success of the reinforcement learning is the competition between agents. This concept is also exploited in Generative Adversarial Networks (GAN) [93], proposed by Ian Goodfellow in 2014. The idea is to promote a competition between a network able to generate samples (generator) and another network (discriminator) able to discriminate whether such samples are original or drawn by the DNN. This method has recently succeeded to generate natural data samples of different nature, including images [93], videos [278], text [205], and speech [272].

The recent progress is also fostered by the numerous toolkits available to the research community, such as Theano [267], TensorFlow [68], CNTK [252], MXNet [44], Caffe [129], and Torch [48], just to name a few.

2.3.3 Future Challenges

The rapid rise of deep leaning contributed to spread optimism towards this technology. However, despite of such a great enthusiasm, there are still major scientific challenges to address for really reaching higher levels

of artificial intelligence [230].

A noteworthy challenge is the effective use of unsupervised data. The recent progress mostly involved supervised learning, that have contributed to achieve state-of-the-art performance in numerous fields. The goal of unsupervised learning is to understand the world around us by observation, which is an ability largely missing in current supervised DNNs. Unsupervised learning can contribute to integrate a common-sense knowledge, that might lead to a less superficial understanding of the world. Human brain, for instance, infers important evidences only through observations: kids are able to predict the trajectory of an object falling down even without knowing differential equations or the newton's law. Future unsupervised learning should be able to properly guess properties, correlations and connections across concepts without labeled data. Data annotation requires huge human efforts, and the study of proper techniques to use unlabelled data, could also pave the way to the development of intelligent machines without human efforts. In order to understand the world around us, another possibility is to learn by interactions. It would be thus of great interest the evolution of current reinforcement learning techniques, possibly with a proper integration of the three basic learning a modalities (i.e, supervised, unsupervised, reinforced) in a unified learning framework.

Another challenge concerns the learning efficiency. Current solutions, in fact, require much more information than humans to learn. For instance, humans are able to learn a simple concept (such as a cat) with few examples, while a machines can model it only with hundreds or even thousands of samples. In the future, the development of learning algorithms able to better disentangle and model the factors of variability will be of primary interest.

Moreover, current deep neural networks are able to solve rather efficiently only single tasks that are generally rather limited and specific (like

recognizing faces, classifying sounds, playing Atari). Differently to human brain, current technology is not able to simultaneously solve very different problems at the same time. It would be thus of great interest the development of more effective multi-task strategies capable of better mimic the human brain.

Another challenge for the future is long-life learning. Current systems are first trained and later tested. The idea of long-life learning is to build a never-ending learning system, that continues to learn from the experience and progressively improves its performance.

Beyond the other challenges, a major achievement would also be the development of a kind of "theory of intelligence" that can better steer the development of intelligent machines. The research in the field, in fact, is now solely based on empirical attempts, that are mostly guided by human intuitions. We can compare the current situation of AI with the first attempts done by the Wright brothers to build a "flying machine". Their approach was only based on a trial-and-error strategy, without any notion about the physics of aerodynamics, whose knowledge is clearly very helpful to design modern aircrafts.

Chapter 3

Distant Speech Recognition

Recognizing distant speech is a difficult problem, especially in challenging acoustic environments characterized by significant levels of noise and reverberation. Despite the noteworthy progress of the last years, DSR is still a very active research field, since a natural and flexible speech interaction with a distant machine remains far from being achieved.

The complexity of this problem normally implies the adoption of complex systems. As shown in Figure 3.1, state-of-the-art DSR systems are often based on a combination of different modules that have to properly work together [295]. The signals recorded by distant microphone arrays can be processed by a front-end [29], that could be composed of both a speech enhancement [14] and an acoustic scene analysis module. The role of acoustic scene analysis is to provide useful information about the acoustic scenario, while the speech enhancement has the primary goal of improving the quality of the signal recorded by the microphones. The resulting enhanced signal feeds a speech recognizer, that tries to identify the sequence of words uttered by the target speaker. Note that the scheme of Figure 3.1 represents only one possible implementation of a state-of-the-art DSR system. The order of the components, for instance, might change depending on the specific architecture. For example, the acoustic scene analysis can

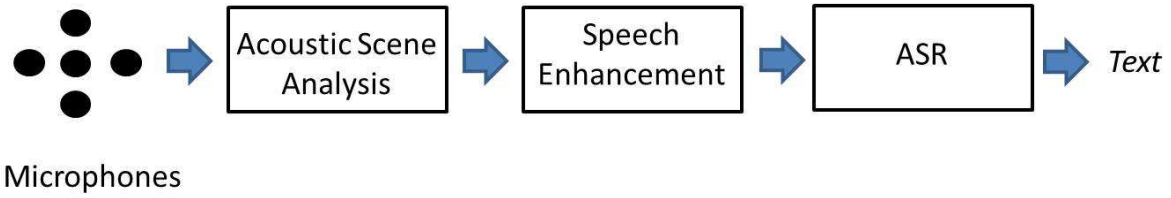


Figure 3.1: An example of distant-talking interaction with a state-of-the-art speech recognition system.

be performed also after the speech enhancement or, in many cases, can be performed both after and before enhancing the speech.

The following sections will first describe in detail the aforementioned technologies. First, the following section proposes a more detailed description of the main challenges in DSR. After that, a summary of the main state-of-the-art technologies for speech reognition, speech enhancement, and acoustic scene analysis is reported in Sec. 3.2, 3.3, and 3.4, respectively.

3.1 Main challenges

The signal $y[n]$ recorded by a distant microphone is described by the following equation:

$$y[n] = x[n] * h[n] + v[n] \tag{3.1}$$

The original close-talking speech signal $x[n]$ (i.e, the speech signal before its propagation in the acoustic environment, that is assumed to be a latent variable not directly observed) is reflected many times by the walls, the floor and the ceiling as well as by the objects within the acoustic environment, as shown in Figure 3.2.

Such a multi-path propagation, known as reverberation [147], is represented by a function called impulse response $h[n]$, that is convoluted

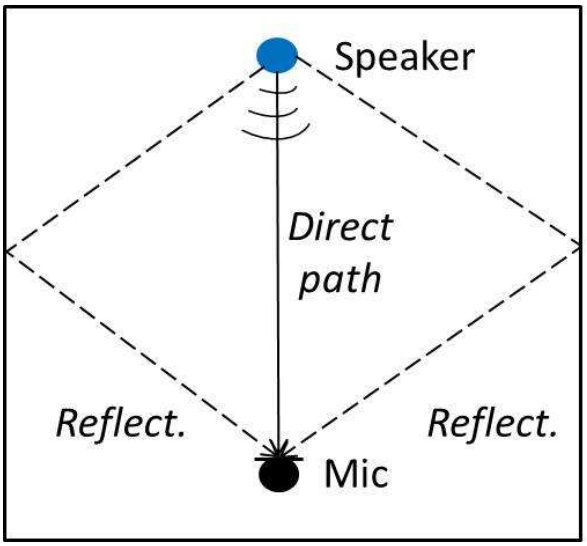


Figure 3.2: Acoustic reverberation in a typical enclosure.

with $x[n]$. The recorded signal $y[n]$ also includes the contributions $v[n]$ of competitive sources, such as other speakers, telephone ringing, music and other possible interfering background noises. Figure 3.3 shows an example of close-talking speech $x[n]$ with a corresponding distant-talking signal $y[n]$ corrupted by both noise and reverberations, highlighting the deleterious effects of these disturbances. In the following sub-sections, the main characteristics of noise and reverberation in a real DSR application are better discussed.

3.1.1 Acoustic reverberation

The impulse response $h[n]$ (see Figure 3.4) can be modeled as a long and causal FIR filter (i.e., $h[n] = 0 \quad \forall n < 0$), whose taps describe the propagation of the signal in the environment. In particular, if one assumes to deal with a linear time-invariant acoustical transmission system, the impulse response provides a complete description of the changes a sound signal undergoes when it travels from a particular position in space to a given microphone [147].

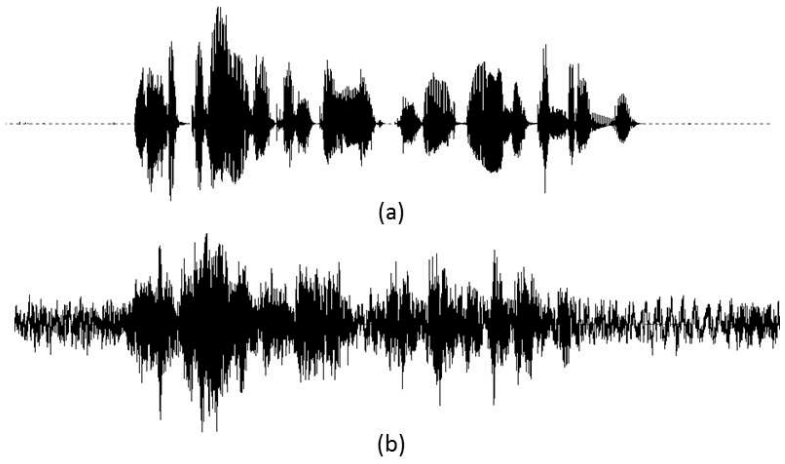
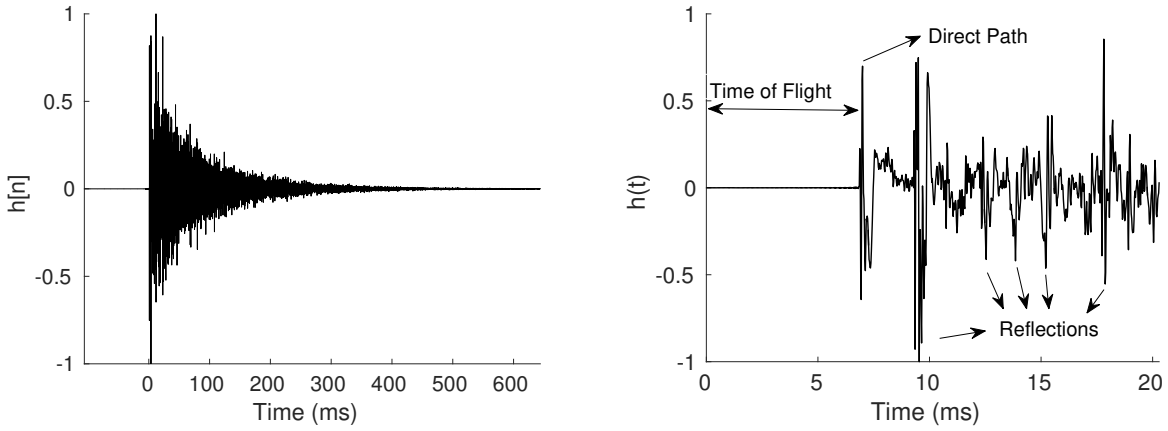


Figure 3.3: An example of close-talking speech $x[n]$ is depicted in (a), while the same sentence $y[n]$ recorded with a distant microphone in a noisy and reverberant environment is shown in (b).



(a) Impulse Response $h[n]$

(b) Zoom on the first reflections of $h[n]$

Figure 3.4: An impulse response measured in a domestic environment with a reverberation time of about 780 ms.

An impulse response of a typical enclosure is composed of three basic components: the direct sound, early reflections and late reverberation. The direct sound travels from the sound source to the microphone along a straight line. The time taken by the direct sound to reach the microphone is called Time of Flight (ToF). The early reflections, that arrive at the mi-

crophone within approximately 50-100 ms after the direct sound, are a set of discrete replicas deriving from various reflecting surfaces, such as walls, floor and ceiling. Finally, late reflections are based on a dense succession of echoes, where individual contributions can no longer be discriminated. Such late echoes are more dense in time but characterized by a diminishing energy, due to the attenuation derived from longer paths. The combination of these three contributions originates the typical exponential decay of the impulse response, that can be appreciated in Figure 3.4a.

Acoustic reverberation significantly impairs the intelligibility and the quality of a speech signal, that become more difficult to interpret by both humans and, even to a greater extent, by automatic speech recognizers. Although the global effects of this disturbance can be summarized with a linear filter, modeling the effects of reverberation in a statistical way or even removing it for the recorded signal is difficult for several reasons. First of all, reverberation introduces long-term effects on the speech signal, originating impulse responses with a large number of filter coefficients, that are difficult to be precisely estimated in a blind fashion. Secondly, the impulse response is a non-stationary function that might change substantially depending on several factors, including the room geometry, the presence of objects in the acoustic environment, position, source/microphone directivity, source/microphone frequency response, temperature, humidity, and air flow, just to name a few.

The amount of reverberation characterizing an acoustic enclosure can be estimated with some popular metrics. One of the most representative measures is the reverberation time T_{60}, that is defined as the time required for a sound to decay 60 dB from its initial energy level. Although T_{60} varies significantly depending on the room acoustics, in standard enclosures it typically ranges between 250 ms and 850 ms. Figure 3.5 shows the log-energy decay obtained with the impulse response of Figure 3.4. In this

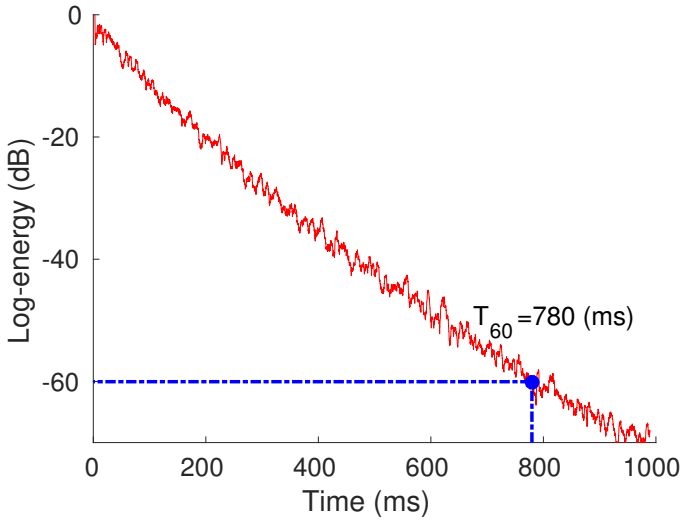


Figure 3.5: Log-energy decay of $h[n]$

case, the energy decay of $h[n]$ indicates that T_{60} is about 780 ms. Another important measure is the Direct-to-Reverberant Ratio (DRR), that is the energy ratio between the direct and reverberant components of the impulse response.

A direct measurement of the room characteristics starting from distant-talking speech signals is normally not possible, and several solutions have been proposed in the literature to blindly estimate both T_{60} and DRR [62]. The estimation of these parameters can be very useful for DSR systems, since it provides valuable information to speech recognition and enhancement algorithms. For instance, these measures can be used to inform dereverberation algorithms, to select a proper microphone in the environment or to choose a suitable acoustic model, as will be discussed in the following of the thesis.

3.1.2 Additive noise

The recorded signal $y[n]$ also includes the contributions $v[n]$ of additive interfering noises (e.g., music, fans, other speakers).

Additive noise can be classified into different categories. According to its time evolution, the noise can be continuous (e.g., fans), intermittent (e.g, telephone ringing) or impulsive (e.g, a door knock). It can also be classified according to its frequency characteristics (white, pink, blue, etc.) or according to its statistical properties (gaussian, poisson, etc.). Moreover, the noise can be stationary or non-stationary, depending on the time evolution of its statistical features. In a realistic environment, $v[n]$ might be composed of several noise sources of different types that are simultaneously active, making this term very difficult to model or remove from $y[n]$.

The amount of noise is normally measured with the Signal-to-Noise Ratio (SNR). Similarly to reverberation-related measures, several approaches have been proposed to estimate SNR [290], and its knowledge can be useful at the different stages of a distant speech recognition system.

3.2 Speech Recognition

Speech recognition represents the core technology of a DSR system. Its goal is to convert a speech signal (possibly processed by an enhancement module) into the sequence of words uttered by the speaker. As shown in Figure 3.6, the speech recognizer is composed of several interconnected modules, that performs feature extraction, acoustic and language modeling as well decoding. The following sub-sections will describe these basic components of ASR. The section will then continue with an overview of the most popular techniques for achieving robustness (Sec. 3.2.6) and with a discussion of modern end-to-end systems (Sec. 3.2.7). Finally, a summary of the ASR history is proposed in Sec. 3.2.8.

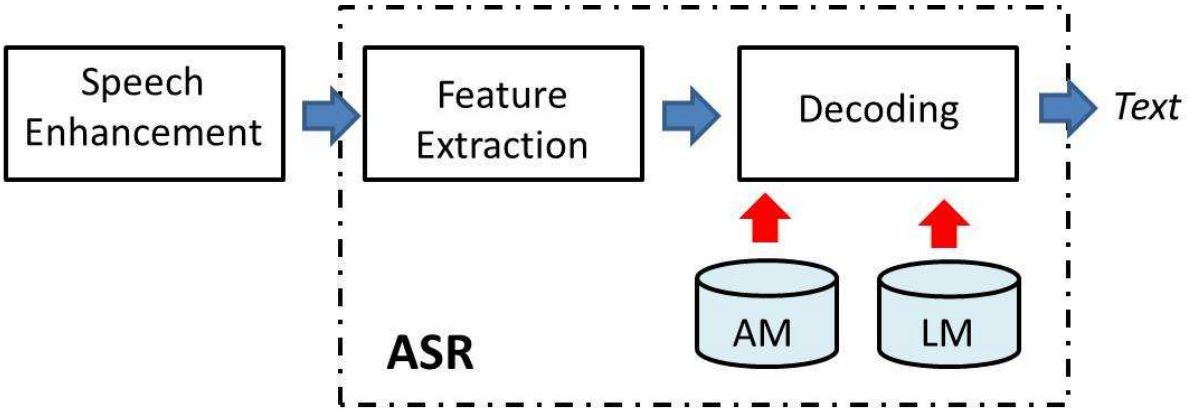


Figure 3.6: A block diagram highlighting the main components involved in a speech recognizer.

3.2.1 Problem Formulation

The problem of speech recognition consists in finding the most likely sequence of words $\hat{w}$ given the features y extracted from the signal $y[n]$ [290]. More formally, the problem can be formulated using the maximum a posteriori (MAP) estimate as follows:

$$\hat{w} = \arg\max_{w} P(w/y) \tag{3.2}$$

using the Bayes' theorem, the previous equation can be rewritten as:

$$\hat{w} = \arg\max_{w} \frac{P(y/w) \cdot P(w)}{P(y)} \tag{3.3}$$

$$= \arg\max_{w} P(y/w) \cdot P(w) \tag{3.4}$$

The likelihood $P(y/w)$ is determined by an acoustic model, while $P(w)$ is determined by a language model. The information provided by these models is integrated in a search graph, that is decoded to estimate the sequence of words uttered by the speaker.

In the following, the individual components of a speech recognizer will be discussed.

3.2.2 Feature Extraction

Feature extraction aims to represent the speech waveform $y[n]$ with a reduced set of parameters y, while preserving most of the information needed to discriminate the spoken units and the speech characteristics. Although many different features were proposed in the literature, the most popular are the Mel-Cepstral Coefficients (MFCCs) [57], that attempt to incorporate concepts from the human auditory processing and perception. MFCCs are commonly derived by processing the speech signal in the following way:

- Split the signal into frames (usually of 20-25 ms with 10 ms of overlap) using a windowing function (e.g., a Hamming window).
- Compute the spectrum of each frame with the Fourier transform.
- Apply the mel-filterbank to the power spectra. A mel filterbank consists of triangular overlapping windows that are spread over the whole frequency range. To mimic the non-linear human ear perception of sound, these filters are more discriminative at lower frequencies and less discriminative at higher frequencies.
- Take the logs of each of filter output. The features processed in this way are called in the literature FBANKs and are often used as input parameters for DNNs.
- Take the discrete cosine transform of the FBANK features.
- Select a subset (usually 13) of the transformed features.

An example of MFCC features is reported in Figure 3.7. Several alternatives have been proposed in the literature [165], including Perceptual linear prediction (PLP) [113] and biologically-inspired spectro-temporal features such as Gabor [139, 174] or gammatone features [246]. Pitch and

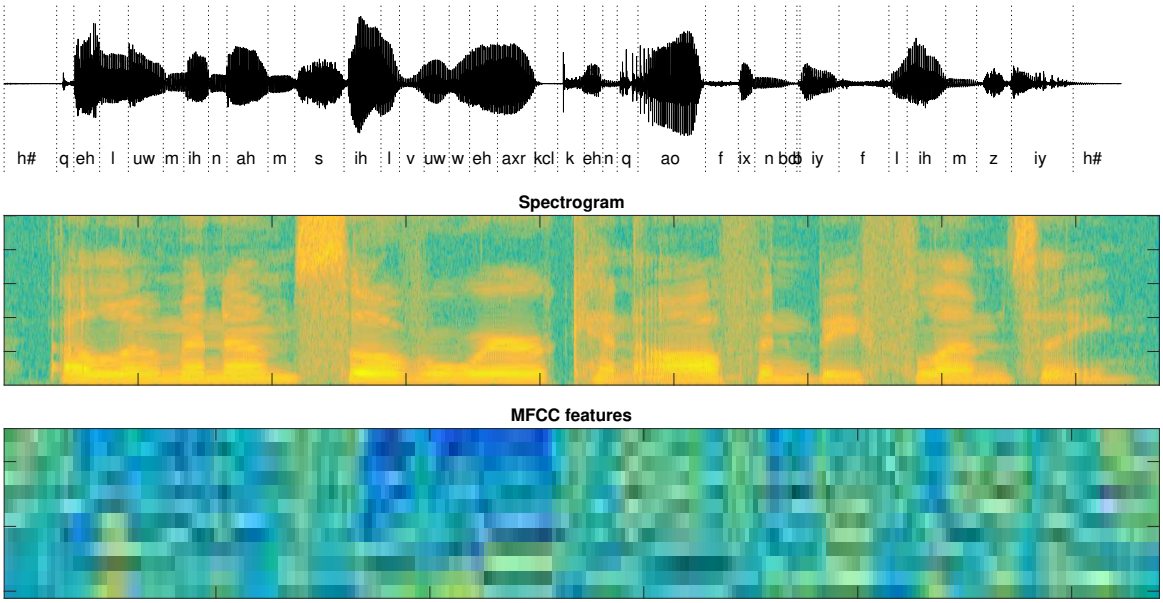


Figure 3.7: Spectrogram and MFCC features (without log-energy) for a phonetically-rich sentence of the TIMIT dataset.

Probability of Voicing (PoV) are also often used as additional parameters [153, 194, 86], that resulted helpful for improving the ASR performance.

Ideally, acoustic features used for ASR should be independent of the specific characteristics of a given speaker. With this goal, several feature normalization techniques such as Cepstral Mean and Variance Normalisation (CMVN), Vocal Tract Length Normalization (VTLN) [64] or Feature space Maximum Likelihood Linear Regression (fMLLR) [260] have been proposed. A powerful approach to obtain robust models is to combine standard ASR features with i-vectors, that can be used to inform the ASR system about speaker identity or environmental characteristics [136]. Beyond speaker normalization purposes, a feature transformation can be employed to perform dimensionality reduction or to decorrelate the information. Examples of these transformations are LDA, HLDA [259] and PCA. More sophisticated solutions are based on non-linear transformations achieved by neural networks using, for instance, TANDEM [114]

and Bottleneck architectures [104].

As outlined before, several research efforts have been devoted in the past to properly design robust acoustic features for ASR [7]. Deep learning, however, is drastically changing the way feature extraction is approached [297]. In the context of deep learning, in fact, the acoustic features can be automatically learned from data, without the need of human efforts and hand-crafted features. Differently to past ASR systems, the trend is thus to feed the neural network with simple low-level speech representations (such as FBANKs features), leaving the DNN to freely extract higher level parameters. Some noteworthy attempts have also shown that it is possible to directly feed a neural network with raw speech samples [195, 241, 242]. These recent works have consistently shown that the first layer of a CNN is able to derive a set of spectro-temporal filters similar to that used for Gabor feature extraction.

3.2.3 Acoustic Model

The objective of the acoustic model is to provide a statistical representation of the basic sounds making up speech, which is inferred by properly analyzing many hours of recordings with their corresponding text transcriptions. Modern speech recognizers use more than 1000 hours of annotated speech for training the acoustic models [4]. As anticipated in the previous sections, state-of-the-art acoustic models are based on Hidden Markov Models (HMMs) [208].

HMMs are a popular generative models for representing probability distributions over sequences of observations, that provide a rather straightforward way to model speech. Within the HMM framework, the basic sounds of the speech can be modeled by a set of simple HMMs. As shown in Figure 3.8, each basic HMM can describe a particular phone and is generally based on a short sequence of states (normally three or five), that might

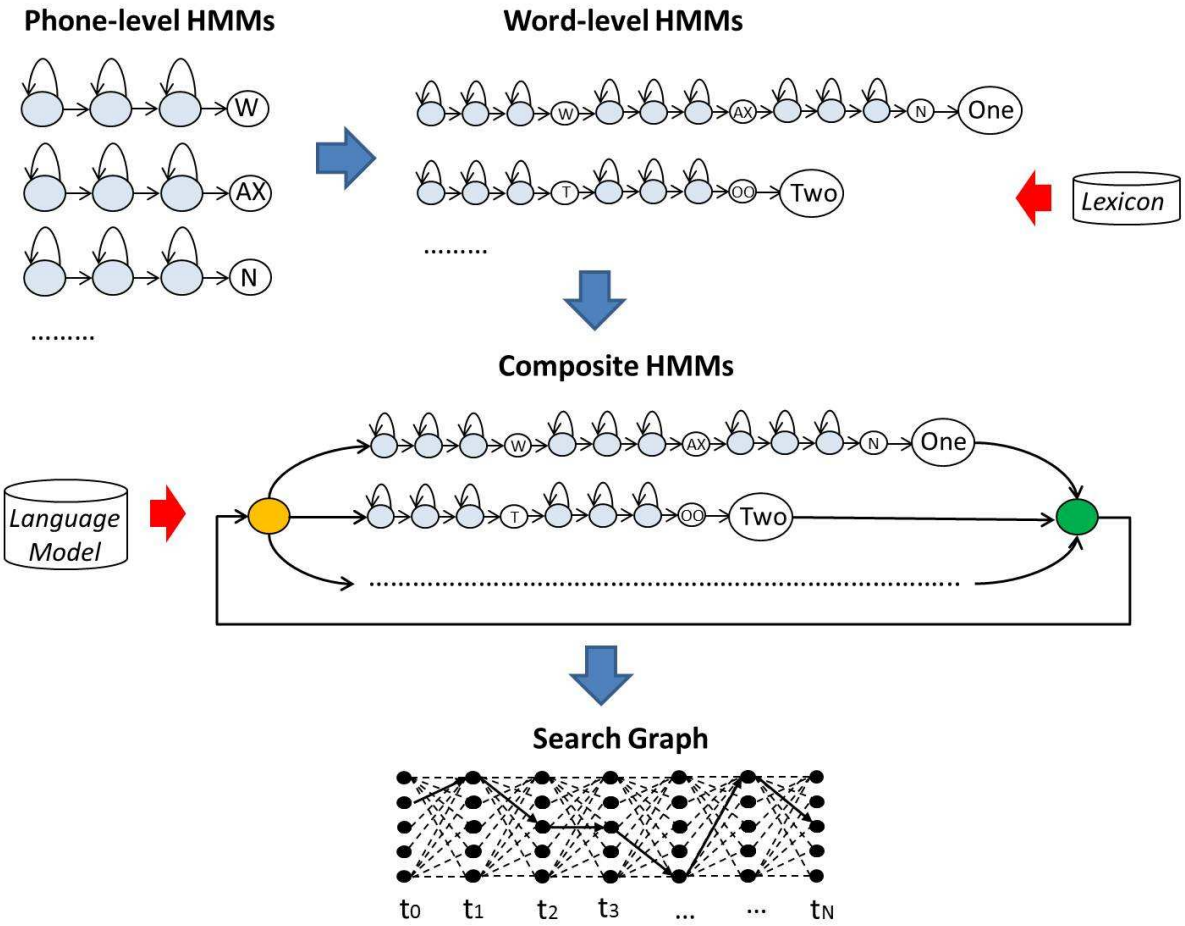


Figure 3.8: An example of Hidden Markov Models (HMMs) for speech recognition.

represent the evolution in time of the considered phoneme. This allows the acoustic model to separately model beginning, central and last parts of each basic sound. Each model is characterized by observation and transition probabilities. The transition probabilities model the duration of each state, while observation probabilities describe how likely a certain phone unit has been generated by the current HMM state.

Modern acoustic models actually operate at context-dependent level, where instead of using phones as basic speech units, more articulated fundamental sounds considering also the surrounding phones are adopted. The definition of proper context-dependent phone states is normally based on

a Phonetic Decision Tree (PDT), that clusters phones based on a priori linguistic knowledge and/or exploiting acoustic similarities observed from data [121].

These basic HMMs are the building blocks for progressively composing more complex models. As shown in Figure 3.8, word-level HMMs can be derived by concatenating phone-level HMMs. This operation requires a lexicon, that converts each word w into a corresponding sequence of phones. Although for some languages (e.g., Italian) this conversion is rather straightforward, for many others (e.g, English) it might be vary tricky and requires sophisticated techniques as well as human efforts to derive an accurate lexicon [2].

Word-based HMMs can be, in turn, combined to generate a composite HMM, that can model any sequence of words in the vocabulary thanks to the feedback connections. As will be discussed in Sec. 3.2.4, the composite HMM can exploit a language model to attribute more weight to the paths leading to likely sequence of words. The obtained HMM is then optimized (by removing, for instance, redundant states) and, starting from it, a search graph is derived to estimate the most likely sequence of words uttered by the speaker, as will be discussed in Sec. 3.2.5.

Depending on the paradigm used for computing the observation probabilities, we can distinguish between GMM or DNN-HMMs. Modern speech recognizers are based on the latter framework and estimate these probabilities with a DNN. DNN-HMM speech recognizers are often called *hybrid* systems, since they are based on both a generative (HMM) and a discriminative (DNN) model, offering a significant performance gain over GMM-HMM solutions due to the following reasons:

- **Better generalization**; DNNs can generalize much better, since they jointly model distributions of different classes. In GMMs, each state is modeled separately with an independent set of GMMs, limiting the

Algorithm 1 Kaldi s5 Recipe for DNN Training

1: Train a GMM-HMM monophone model (*mono*)
2: Align the training sentences with the *mono* model
3: Derive a Phonetic Decision Tree and define the context-dependent states s
4: Initialize the new model with the mono model
5: Train a GMM-HMM triphone model (*tri1*)
6: Align the training sentences with the *tri1* model
7: Initialize the new model with the *tri1* model
8: Train a GMM-HMM triphone model with LDA (*tri2*)
9: Align the training sentences with the *tri2* model
10: Initialize the new model with the *tri3* model
11: Train a GMM-HMM triphone model with LDA+fMLLr (*tri3*)
12: Align the training sentences with the *tri3* model to obtain DNN labels
13: Train a DNN-HMM

possibilities to share correlation across similar states.

- **Ability to manage long contexts**; Differently to GMMs, that can effectively handle feature spaces with a reduced dimensionality, DNNs can naturally manage large input spaces. This allows the DNN to account for longer time contexts, that are crucial to improve the system performance.

- **Ability to exploit correlated features**; Within the HMM-GMM framework, diagonal covariance matrices are often considered to train a compact model. However, this assumption forces the system to adopt uncorrelated features (such as MFCCs), since correlation across different dimensions cannot be modeled with simple diagonal covariance matrices. Differently, DNNs are also compatible with correlated inputs and offer the flexibility to effectively exploit features of different nature (such a i-vectors and FBANK features).

The training of a GMM-HMM ASR system is carried out with the Baum-Welch algorithm [12], that is an example of expectation-maximisation

(EM) approach [59]. In the context of modern DNN-HMM systems, instead, the training pipeline is more complex and usually requires the prior training of a GMM-HMM model for deriving the labels needed for DNN training. Such a complexity can be clearly noticed in the standard Kaldi recipe [204] reported in Alg. 3, where a large number of incremental steps are needed to train a state-of-the art speech recognizer.

3.2.4 Language Model

The language model $P(w)$ is learned by means of large text corpora and has the key goal of considering that some sequences of words (e.g., "open the door") are more likely than others (e.g., "open the dog"). The most popular language models are based on n-grams, which gather statistics by counting the occurrences of contiguous sequences of n words. Typically, this statistics are adjusted by smoothing techniques, such as the Kneser-Ney method [141]. Depending on the number n of previous words considered to estimate the current one, bigram, trigram, four-gram or even five-grams LMs can be trained.

An alternative to n-gram language models is offered by neural language models [17, 176], that learn distributed word representations to counteract the curse of dimensionality problem. Despite the effectiveness of neural language models, their efficient integration into the search algorithm is still an open issue. Due to this limitation, neural language models can effectively be used only for lattice rescoring after a first decoding step, as discuss more in detail in the following subsection.

3.2.5 Search & Decoding

Once the acoustic and language models are trained, the decoding procedure allows the speech recognizer to estimate the sequence of words $\hat{w}$ uttered

by the speaker. The information embedded in the acoustic and language models are integrated in a search graph, where each path starting from the initial to the final state represents a particular sequence of words [121]. Among all the possible alternatives, the goal of the decoder is to find the most likely one, that corresponds to the most probable sequence of words.

A naive solution would be to search for all the possible paths and choose the one with the highest likelihood. This approach, however, is unfeasible especially for large vocabulary ASR. For a vocabulary of size V words and a sequence of M words, in fact, there are V^M alternatives to evaluate. For this reason, search is typically based on the Viterbi algorithm, that is an efficient recursive algorithm based on dynamic programming [277]. The Viterbi algorithm is able to perform an optimal exhaustive search and only needs to memorize the most probable path at each state, significantly saving both computations and memory. For large vocabulary applications, however, even the straightforward application of Viterbi search leads to an excessive computational complexity [208].

To make decoding computationally tractable, beamsearch methods are often applied [121], which define some heuristics to prune non promising paths. Another way to make search more efficient is to save computations by sharing word prefixes inside the search graph. Multi-pass search [80] can also be used for reducing the computational complexity by progressively using more detailed models. For instance, a first decoding step can be performed with a bigram language model, followed by a second pass based on a more precise trigram LM. In the latter case, the first decoding step outputs a list of the N best alternatives, that can be summarized in a word lattice. The word lattice can be efficiently rescored with a more precise language model, that can also be a neural language model based on recurrent neural networks [176]. Modern decoders for speech recognition are implemented with Weighted Finite State Transducers (WFST) [180], that

offer a flexible and efficient framework to implement the aforementioned decoder functionalities.

3.2.6 Robust ASR

The acoustic model is a crucial component of the speech recognizer, whose accuracy has a significant impact on the final ASR performance. The development of effective acoustic models, however, is a challenging problem, due to the required independence from many variability factors, including speaker accents, speaking rates, hesitations and spontaneous speech. In addition to these issues, a DSR system should also be robust against noise and reverberation. As will be discussed in Sec. 3.3, robustness in adverse acoustic environments can be in part achieved with the speech enhancement module. However, since even very advanced speech enhancement solutions are not able to completely neutralize the unwanted disturbances, robustness is also required to the acoustic model of a speech recognizer. The most effective DSR systems, in fact, are based on a robust speech enhancement module that attempts to minimize the harmful effect of noise and reverberation, followed by an acoustic model that tries to statistically model the residual disturbance not removed by the front-end [295].

A number of techniques have been proposed in the literature to derive robust acoustic models. One of the most effective and straightforward approaches consists in training training the ASR system with many data. A popular approach is multi-style training [251, 50, 108], in which the acoustic models are trained with data derived from many environments characterized by different noisy and reverberant conditions. Since the open availability of large corpora is still an issue, popular approaches are based on transforming existing corpora through data augmentation [52, 295, 210, 142] or contaminated speech methods [168, 50, 108, 138, 11, 219, 32], that will be discussed in detail in Chapter 4.

Another popular approach is to improve the robustness of a speech recognizer with acoustic model adaptation, that can be supervised or unsupervised depending on the availability of annotations in the adaptation corpus. In the context of GMM-HMM systems, MAP or MLLR [260] adaptation gained particular attention in the past. For modern DNNs, various techniques have been proposed [178]. The simplest solution is to adapt the DNN model by performing some additional retraining iterations. However, when little adaptation data are available, the latter approach can be prone to overfit the adaptation corpus. To mitigate this issue, a possible solution is to add a regularization term in the DNN cost function, such as the Kullaback-Leiber divergence. The regularizer forces the adapted distribution to stay close to the original one [298]. Particular attention has also been devoted to unsupervised adaptation. Typically, this kind of adaptation is carried out by performing a first decoding step with non-adapted acoustic model, that provides a rough transcriptions of the speech. The obtained transcription is used to derive an adapted acoustic model. The effectiveness of this approach depends on the quality of the initial transcription, that might contain several recognition errors potentially able to impair the effectiveness of the adaptation strategy. An effective approach studied in [70] consists in properly filtering the first-step transcriptions according to sentence-level ASR confidence measures.

Other effective techniques for improving acoustic modeling are the methods for sequence discriminative training, that have the main goal of training the system with metrics like maximum mutual information (MMI), boosted MMI (BMMI), minimum phone error (MPE) or minimum Bayes risk (MBR) that have been proved more robust than traditional ones (e.g., log-likelihood, cross-entropy) [121, 297, 273].

When multiple recognition systems are available, it could also be convenient to combine the hypothesized word outputs and select the best scoring

word sequence. With this goal, a popular technique is ROVER [75], that adopts a voting solution implemented with dynamic programming to produce the final combined output.

3.2.7 Towards end-to-end ASR

As discussed in the previous sections, deep learning has recently contributed to replace GMMs with DNNs inside the HMM framework. Following this trend, it is easy to predict that HMMs will be the next element of the ASR pipeline that will be replaced by deep learning. Despite their effectiveness, HMMs have, in fact, some important limitations [87]. First of all, they rely on the first order Markov assumption, which states that the probability of being in a given state at time t only depends on the state at time $t-1$. This is a strong assumption that doesn't represent well speech signals, whose dependencies might extend through several states. Moreover, HMMs assume that successive observations are independent, while consecutive features are clearly highly correlated in a speech signal.

Modern RNNs promise to overcome these issues. HMM-free speech recognizers have been, in fact, recently proposed under the name of end-to-end systems. End-to-end speech recognizers are not only potentially able to overcome the limitations of HMMs, but also aim to avoid any human effort in the design of the speech recognizer, trying to learn everything in a discriminative way from (large) datasets.

Popular end-to-end techniques are attention models [9] and Connectionist Temporal Classification (CTC) [96]. Attention models are based on an encoder-decoder architecture coupled with an attention mechanism [9] that decides which input information to analyze at each decoding step. CTC [96] is based on a DNN predicting symbols from a predefined alphabet (characters, phones, words) to which an extra unit (*blank*) that emits no labels is added. Similarly to HMMs, dynamic programming is used to sum

over all the paths that are possible realizations of the ground-truth label sequence to compute its likelihood and the corresponding gradient with respect to the neural network parameters. This way, CTC allows one to optimize the likelihood of the desired output sequence directly, without the need for an explicit label alignment.

Despite some promising results recently achieved with huge dataset by Baidu [4], the ASR performance of end-to-end systems is generally worse than that currently achievable with state-of-the-art hybrid systems. The former methods, in fact, are relatively young models, and still need to progressively improve to really compete with a more mature DNN-HMM hybrid technology [175].

3.2.8 Brief History

The idea of building machines able to recognize speech has fascinated people for long time. The appeal of speech technologies is, after all, also witnessed by the numerous movies, such as "*2001: A Space Odyssey* and the "*Star Wars*" saga, showing advanced robots or computer naturally interacting with human beings.

Researchers started addressing the problem of speech recognition in the early 50s. The first attempts were based on template matching approaches, whose core idea is to compare a low-level speech representations with a set of predefined patterns. An example is the pioneering system developed in 1952 by Davis, Biddulph, and Balashek at Bell Laboratories [132]. The system was able to recognize isolated digits from a single speaker, using the formant frequencies computed during vowel regions. The obtained formant trajectories served as the reference pattern for determining the identity of an unknown digit. Another system using a similar technology was described in [190], where a speech recognizer able to classify 10 syllables of a single talker was proposed. The work described in [77] was, instead,

the first speaker-independent vowel recognizer.

In the 60s, the concept of adopting non-uniform time scale for aligning speech patterns started to gain interest. For instance, Vintsyuk [276] proposed the use of dynamic programming for deriving more robust similarity measures using time alignment between two utterances. This initial work was followed by the studies of Sakoe and Chiba [245], that proposed more formal methods, known as dynamic time warping, for speech pattern matching.

In the early 70s, a significant achievement was the development of Linear Predictive Coding (LPC) [125], that resulted important to derive robust features for recognition performance. Another important milestone was the development of the Hearsay system by Carnegie Mellon University (CMU), that was the first system exploiting beam search for speech recognition.

During 80s, there was a progressive decline of pattern matching methods in favor of more robust statistical approaches [209], that were based on a statistical description of the speech signal at both acoustic and linguistic levels. This trend, initially promoted by both Bell Labs and IBM, is well summarized by the Mercer's famous comment: "*There is no data like more data*", or by the Fred Jelinek's aphorisms : "*Every time I fire a linguist, the performance of the speech recognizer goes up.*". Hidden Markov Models (HMMs), studied by Baum at the Princeton University in the early 70s [12], emerged as the dominant paradigm for statistical ASR. HMMs, which are still used in state-of-the-art speech recognizers, drastically revolutionize ASR due to some important peculiarities [208], including the availability of efficient training and inference algorithms based on dynamic programming [12], the flexibility in merging statistical information deriving from acoustic, lexicon and language models, as well as the robustness in handling acoustic variabilities of speech.

In the 90s, the progressive evolution of this framework, led to a gradual

maturation of the technology [121], which in some ways lasts to this day. The main achievements were the evolution from discrete to continuous HMMs as well as the development of context-dependent speech recognizers able to significantly outperform context-dependent HMMs [121].

For more than three decades, the dominant approach was based on HMMs coupled with Mixture Gaussian Models (GMMs). Starting from 2012, the rise of deep learning started to revolutionized speech recognition [297] by replacing GMMs with DNNs [55]. DNN-HMMs speech recognizers contributed to significantly outperform previous systems, laying the foundations for a new generation of speech recognizers. Actually, the first attempts to exploit the discriminative power of DNNs in ASR were done in the early 90's by Boulard and Morgan [27], following the connectionism trend introduced in the deep learning chapter. Such early works were promising but rather premature, since at that time hardware, data and algorithms were not as mature as today.

The progress in ASR, however, was also fostered in last decades by several other factors, including the public release of speech corpora [82, 198, 90, 5] and the development of ASR toolkits [296, 204]. The numerous evaluation campaigns promoted by DARPA and NIST, as well as the challenges such as CHiME [11, 10] and REVERB [138] were also of fundamental importance to promote ASR progress and to establish common evaluation frameworks across researchers. Important contributions were also given by some speech-related European Projects, such as AMI/AMIDA [227], DICIT [191] and DIRHA. Last but not least, the renewed interest in ASR has recently encouraged huge investments in the field by several big companies (such as Google, Microsoft, IBM, Amazon, Nuance, Baidu, Apple), that in many cases actively contribute to the basic research in the field.

3.3 Speech Enhancement

Speech enhancement to counteract the effects produced by environmental noise and reverberation was studied for decades, targeting not only speech recognition but also different application fields, ranging from hearing aids to hands-free teleconferencing. Many methods are described in the related literature both for single and for multi-channel input [126, 14]. Common speech enhancement has the primary goal of improving the quality at perceptual level of the recorded signal $y[n]$, trying to minimize the deleterious effects of noises $v[n]$ and reverberation $h[n]$. In the following sections, the most popular speech enhancement techniques (e.g., spatial filtering, spectral subtraction, DNN-based speech enhancement) and problems (e.g., dereverberation, source separation, AEC, microphone selection) are discussed.

3.3.1 Spatial Filtering

One of the most effective multi-microphone technique of speech enhancement is spatial filtering or beamforming [135]. The goal of these methods is to obtain spatial selectivity (i.e., privilege the areas where a target user is actually speaking), limiting the effects of both noise $v[n]$ and reverberation $h[n]$.

A straightforward way to perform spatial filtering is provided by the delay-and-sum beamforming, that simply performs a time alignment followed by a sum of the recorded signals. The time alignment, that is necessary since the target signal reaches the microphones at different time instances, is generally achieved by computing delays with TDOA techniques [140]. A useful tool to analyze the directional properties of a spatial filtering algorithm is the polar pattern, that highlights the sensitivity of the beamformer for all the possible angles from which the sound might

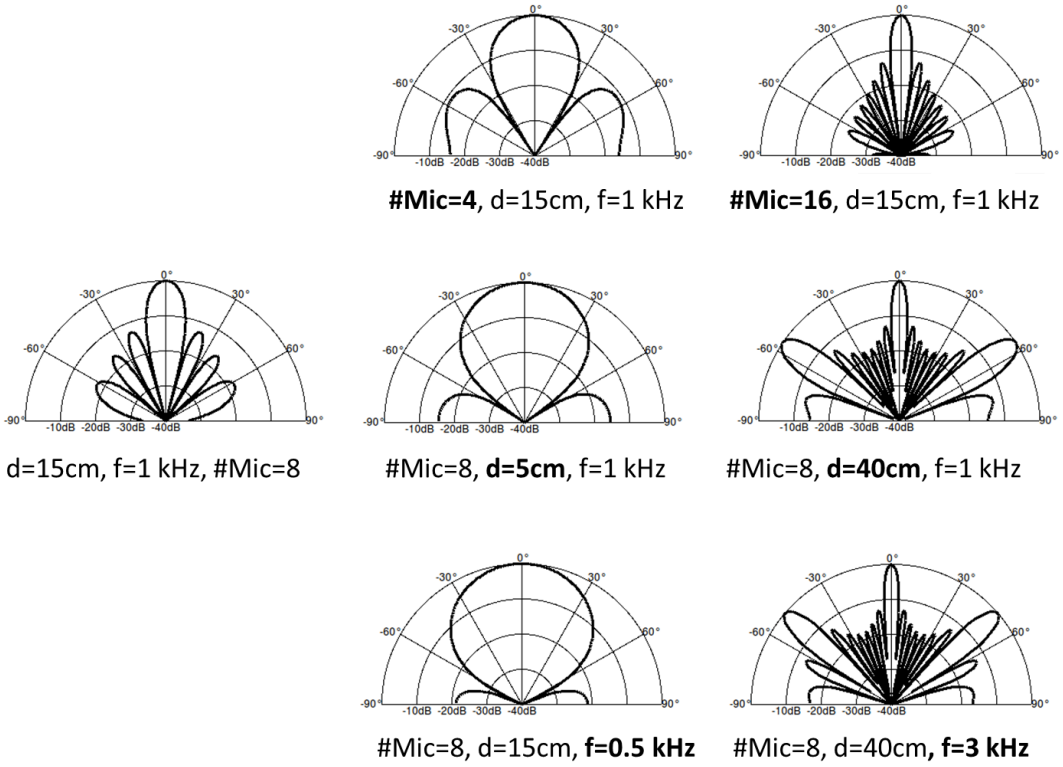


Figure 3.9: Examples of beampatterns obtained when delay-and-sum beamforming is applied to different linear arrays characterized by different microphone configurations.

arrive.

Examples of polar patterns obtained with a delay-and-sum beamforming applied to a linear array are depicted in Figure 3.9. The figure shows that the number of microphones $\#Mic$, their spacing d as well as the frequency f of the considered signal have an impact on the resulting directivity. In particular, the main lobe tends to become sharper as the number of microphones, the sensor spacing d, and the frequency f of the incident signal increase. The microphone spacing d, however, should be carefully designed to avoid *spatial aliasing* problems. A linear array, indeed, can detect a plane wave signal at wrong angles if the signal wavelengths are shorter than twice the distance d. The spatial aliasing phenomenon causes the appearance of *grating lobes* in the delay-and-sum beampattern, that

are highlighted in Figure 3.9 when d and f assume high values.

Despite its simplicity, delay-and-sum beamforming has some drawbacks. For instance, as shown in Figure 3.9, it actually generates proper directional patterns only for some specific subbands, and it tends to be less directional at lower frequency. Moreover, it introduces spatial aliasing at higher frequency bands.

To mitigate these issues, a more advanced solutions is filter-and-sum beamforming [133], that filters the recorded signals before the time alignment. Another alternative is represented by super-directive beamforming [24], which further enhances the target speech by suppressing the contributions of the noise sources from other directions.

Another limitation of current beamformers is that they usually rely on the assumption that the target speaker is always the dominant source in the environment. Although this can occur in rather quiet environments, such an assumption could not hold anymore in very noisy situations, where the targeted speaker may be weaker than competitive sources.

3.3.2 Spectral Subtraction

Differently from beamforming, that aims to counteract the effects of both noise and reverberation, some other approaches proposed in the literature try to mitigate just one of these unwanted phenomena. To counteract additive noise, for instance, a popular approach is represented by spectral subtraction algorithms [65], that were one of the first algorithms historically proposed for the enhancement of single channel speech. These techniques reduce the contributions of $v[n]$ by estimating the noise signals when the speaker is not active. The estimated noise spectrum is then subtracted when speech is produced. The main issue concerning these methods is that such techniques are based on the underlying assumption that the additive noise is almost stationary, which is very unrealistic in practical applications.

As a result, the processed speech signal is characterized by significant distortions (known as residual or musical noise), that may have an impact on both the signal intelligibility and the speech recognition performance. A difference with beamforming is that most of the spectral subtraction techniques can also operate on a single channel.

3.3.3 DNN-based Speech Enhancement

The recent rise of deep learning has laid the foundations for the development of speech enhancement completely based on DNNs. The main difference with the traditional signal processing-based techniques described in previous sub-sections is that the complex non-linear function able to enhance the corrupted speech is not designed by humans, but completely learned from data.

In the last years, several approaches have been proposed in the literature for dereverberation (i.e., reducing the effect of $h[n]$) [292], denoising (i.e., limiting the effect of $v[n]$) [163, 159], and for jointly address both issues [73]. The most popular architectures adopted in recent works were feed-forward [293, 294, 146, 270] and recurrent neural networks [294, 143]. The development of DNN-based speech enhancement has also facilitated an effective integration between this technology with the speech recognition module, as will be better discussed in Chapter 6.

Although DNN-based speech enhancement represents a promising research direction, current approaches typically achieve a performance that is still far from the best signal processing-based techniques, especially when addressing both noise and reverberation at the same time. Even though huge corpora reflecting different types of acoustic conditions can be simulated, a prominent issue concerns the lack of generalization of the former systems, that tend to work well only when test conditions similar to that addressed during training occur. This issue reflects the inherent difficulty

to statistically model the huge variability of noises that might be faced in a realistic environment.

3.3.4 Speech Dereverberation

Speech dereverberation methods are signal processing techniques that aim to limit the harmful effects of acoustic reverberation in a distant-talking speech signal [187]. The solutions for reverberation reduction can be divided into many categories (e.g., single vs multiple microphones approaches, feature vs signal domain methods). According to [187], speech dereverberation algorithms can be categorized depending on whether or not the impulse response $h[n]$ needs to be directly estimated.

The first category consists of techniques, known as blind deconvolution or reverberation cancellation methods [110], that are based on an estimate of the impulse response $h[n]$. The estimated impulse response is exploited to reconstruct the anechoic speech $x[n]$ with an inverse-filtering operation.

The second category gathers approaches known as reverberation suppression methods that do not directly require an explicit estimation of $h[n]$, but exploit the characteristics of the speech signal to mitigate reverberation. The so-called statistical methods (that require the availability of models for speech and noise signals) belong, for instance, to this category. See [187] (and the reference therein) for more details about the different approaches proposed in the literature.

In general, speech dereverberation is a particularly challenging problem for several reasons. First, both the speech source and the acoustic channel are unknown, non-stationary and time-varying. As outlined in Sec. 3.1, in fact, the impulse responses depend on several factors, including source-microphone 3-D positions, orientations and polar patterns. Moreover, although reverberation can be described as a linear FIR filter, the IR is very long and it is difficult to approximate it in a very precise way.

Lastly, most of the dereverberation algorithms perform properly in the case of reverberated-only signals, and the presence of additive noise might have a serious impact on the quality of the reconstructed signal.

3.3.5 Source Separation

Sound Source Separation tackles the problem of segregating a target voice from background or competing sources. This scenario, often referred to as *"cocktail-party"*, is frequent in distant-talking scenarios, where the target speech might be overlapped with some interfering noises. The most popular algorithms of source separation are implemented using independent component analysis [166] and most of them are completely unsupervised, avoiding the use of any prior knowledge on the acoustic scene. Despite the noteworthy progresses of the last decade and the considerable success of some international challenges such as CHIME [11], a satisfactory separation is still far from being reached and the enhanced signal can be severely corrupted by artifacts and non-linear distortions.

3.3.6 Acoustic Echo Cancellation

Although the noises $v[n]$ corrupting the target speech are in general almost unpredictable, some of these interferences may be directly acquired at their source. An example is represented by the acoustic signal emitted by a television, which can be captured before its propagation in the acoustic environment. In this situation, Acoustic Echo Cancellation (AEC) techniques have the specific goal of suppressing the known interference from the signals recorded by the distant microphones. The most popular approaches achieve this goal by estimating a set of filter parameters with the Least Mean Square (LMS) algorithm. An example is the Subband Acoustic Echo Cancellation (SAEC) [193], that approaches the problem

using subband-based solution. More recently, a Semi-Blind Source Separation (SBSS) paradigm [188], which includes an a priori knowledge of the known interferer as a constraint in the independent component analysis framework, has successfully been proposed.

3.3.7 Microphone Selection

In the case of distributed microphone networks, instead of combining the information of different microphone signals, the signal with better characteristics is obtained through a microphone selection [288]. These solutions are typically based on distortion measures that aim to rank the channels in a way as close as possible to the unknown performance of the recognizer. Several measures have been proposed in the literature. Examples are scores based on the estimation of the position and the orientation of the speaker [287], solutions based on the estimation of the signal to noise ratio (SNR) [289], and solutions attempting to estimate some features from the impulse responses, such as the direct-to-reverberant ratio [128]. More recently, a microphone selection based on the cepstral distance have been proposed [105]. The main issue, that make channel selection still a challenging research direction, concerns the difficulties in devising distortion measures that are correlated with the ASR performance, especially in real environments characterized by unpredictable noisy and reverberant conditions.

3.4 Acoustic Scene Analysis

Acoustic scene analysis refers to techniques aiming to detect, analyse, and classify the acoustic information diffused in the environment. Acoustic scene analysis is a rather general term that might include a bunch of techniques with several different applications, such as automatic surveillance, smart video conferencing, multi-modal human-computer interaction,

hearing-aids technologies [231, 38]. This term can also have a different shades of meaning depending on the particular area of research. For instance, Auditory Scene Analysis (ASA) or Computational Auditory Scene Analysis (CASA) [30] typically refer to sound segregation based on perceptually meaningful elements.

Even though several other fields can benefit from an acoustic scene analysis, in this thesis such a technology is only intended to provide additional information to possibly help both the speech recognizer and the speech enhancement systems. To this purpose, acoustic event detection, described in Sec. 3.4.1, can be used, for instance, to dynamically select an acoustic model more robust against the detected noise. Similarly, the speech enhancement can improve its performance if a hint on the typology of disturbance affecting the recorded signal is provided. Speaker identification techniques, discussed in Sec. 3.4.1, can be helpful to derive personalized acoustic models, that better match user's voice characteristics. Speaker localization, summarized in Sec. 3.4.3, could be instead very precious to guide beamforming algorithms, in order to create spatial selectivity in the direction from where the user is speaking.

3.4.1 Acoustic Event Detection and Classification

A huge variety of both human and non-human acoustic events can occur in a real situation. Acoustic event detection and classification techniques [266] aim at detecting time boundaries of these sounds and categorizing them into classes.

An important sub-problem consists in solely detecting the speech activity, distinguishing between speech and non-speech classes only. For close-talking purposes, relatively simple speech activity detection techniques based on energy thresholds or zero-crossing rates work reasonably well. For DSR applications, an automatic segmentation of the distant audio signals

into speech and non-speech categories is not only more challenging but also more crucial, since users normally interact with a DSR system in a hands-free modality (i.e., push-to-talk buttons or similar devices are usually not available). Several approaches have been proposed in the literature, including techniques based on periodicity measures [271], statistical models [256] or cross-correlation-based approaches [6]. Recently, deep learning-based speech activity detection gained particular attention, as witnessed by the numerous works published in the literature [69, 74, 239, 122, 268, 301].

Beyond speech activity detection, a richer classification of the acoustic events can be helpful for improving the DSR performance. With this purpose, several efforts have been devoted to acoustic event detection in the context of the European project CHIL, with a particular focus on events that can happen in small environments, like lecture and small-meeting rooms [183, 264]. The Detection and Classification of Acoustic Scenes and Events (DCASE) and TRECVID Multimedia Event Detection challenges further contribute to fostering the progress in the field. More recently, Google [83] publicly released a huge dataset that promises to become a benchmark in the field: it is composed of 632 audio event classes and a collection of more that 2 millions human-labeled sound clips drawn from YouTube videos.

Although several solutions have been proposed, the most popular approaches are based on GMM-HMM [173], Support Vector Machines (SVMs) [265, 305] and, more recently, also on DNNs [84, 218].

3.4.2 Speaker Identification and Verification

Speaker verification consists in validating a user's claimed identity using features extracted from his voice (binary classification). Speaker identification, instead, requires solving a more complex problem, since the system has to explicitly find the right speaker among a set of N alternatives (multi-

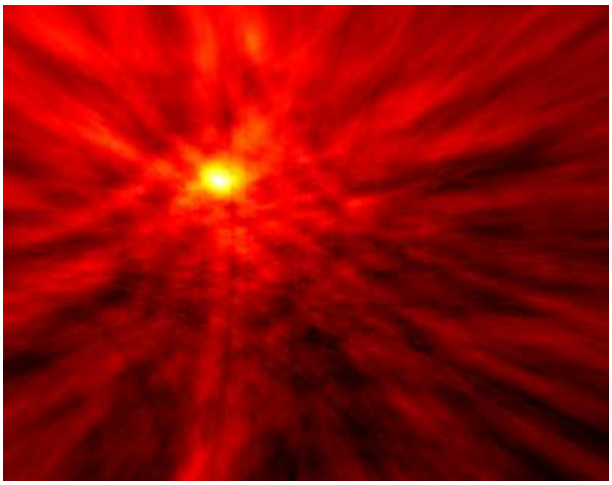


Figure 3.10: An example of GCF obtained with a circular array composed of six microphones. The position of the speaker is clearly highlighted in the map.

class classification).

These technologies can be applied in several fields, including user authentication, surveillance, forensic, security as well as speech recognition. In a command-and-control application, for instance, speaker verification can be used to enable only a particular user to execute speech commands. Speaker ID can be used to derive personalized acoustic models for each different user, allowing the DSR system to significantly improve its performance.

Research in speaker recognition has a long history, dated back to more than half a century ago [78]. Early approaches were based on template matching [206, 207], progressively followed by methods based on HMMs [184], vector quantization [234], GMMs based on Universal Background Model (GMM-UBM) [228] and SVM [248]. State-of-the-art techniques are based on i-vectors [136, 53, 54] and DNNs [229].

3.4.3 Source Localization

Acoustic source localization has the main objective of deriving the spatial coordinates of single or multiple sound sources that are active in a given environment. These technologies have been deeply investigated and several different approaches are available in the literature. In general, popular algorithms are based on the estimation of the Time Differences Of Arrivals (TDOA) at two or more microphones, from which the source location is inferred by applying geometrical considerations. The Generalized Cross-Correlation Phase Transform (GCC-PHAT)[140], is the most common technique for estimating the TDOA at two microphones and relies on phase information only, that turned out to be much more reliable than magnitude for estimating time delays [192]. The TDOA can be exploited to derive acoustic maps, like the Global Coherence Field (GCF) [58], that are particularly effective for accurately localizing acoustic sources as well as to provide important insights on the characteristics of the acoustic environment. An example of a GCF (obtained with a microphone array composed of six microphone) is depicted in Figure 3.10, where the position of the speaker is clearly highlighted. Depending on the microphone configuration, speaker position/orientation as well as on the room acoustic, the GCF function can be affected by several artifacts, including the possible presence of the so-called *ghost peaks*, that might, for instance, appear when strong reflections arise.

In the last decade, the research on speaker localization has led to robust techniques for jointly estimate both speaker position and orientation [34]. Moreover, techniques for simultaneously localizing multiple acoustic sources [35] and methods for speaker tracking [33] have been explored in the literature.

In the context of distant-speech recognition, the localization information

is typically exploited by the speech enhancement system to steer spatial filtering towards the desired acoustic source, as shown in the speech enhancement section.

Chapter 4

Methods for Speech Contamination

A key ingredient for the success of deep learning is the availability of very large corpora, that can be exploited to train deep neural networks with higher robustness and capacity. Nevertheless, the open access of so large annotated datasets is still an issue. Huge speech corpora are, in fact, typically collected by big tech companies, that are reluctant to publicly distribute them. This might create a significant gap between industrial and academic research, eventually hindering the progress of the field.

It is thus of great interest the study of data augmentation approaches [52, 295, 210, 142] that can help academic researchers mitigate these limitations. In the field of speech recognition, data augmentation artificially creates new samples by processing available speech sentences. This can be realized, for instance, by perturbing pitch, formants or other speaker characteristics. The process to transform a close-talking signal into a distant-talking one is often called *data contamination* [168]. Data contamination is typically carried out by convolving close-talking speech recordings $x[n]$ with impulse responses $h[n]$ and adding some noise sequences $v[n]$, as reported in the following equation:

$$y[n] = x[n] * h[n] + v[n] \tag{4.1}$$

The data generated with this approach are often called *simulated data*.

In the literature, some ambiguity between the terms data contamination and multi-style training [251, 50, 108] still exists. In general, the latter refers to training acoustic models with data coming from different domains, regardless whether they are real or contaminated.

This thesis studied crucial aspects behind the data contamination process. We indeed believe that the realism of simulated corpora and, more importantly, the definition of common methodologies, algorithms and good practices to generate such data play a crucial role to foster future research in this field and to eventually help researchers better transfer laboratory results into real application scenarios. Several ingredients are necessary to generate realistic simulated data, including the quality of both the close-talking recordings and the impulse responses. Our studies focused in particular on the latter aspect, considering either IRs measured in the real environment, or derived by room acoustic simulators. The reference environment for the experiments reported in this Chapter was a real apartment available for experiments under the DIRHA project[1]. This apartment, hereinafter referred to as *DIRHA apartment*, was equipped with a microphone network consisting of several microphones. The DIRHA apartment was object of several recording sessions, with the purpose of acquiring real speech datasets and collecting corpora of impulse responses. The reference room for the following experiments was the living-room of the aforementioned apartment, which is characterized by a reverberation time T_{60} of about 750 ms. See the appendix for a more detailed description of the DIRHA apartment (App. A.1, A.2).

The remaining part of this Chapter discusses our main achievements on this topic, summarizing the main findings published in our papers referenced in their respective sections. Sec. 4.1 discusses the proposed methodology to generate high-quality simulated data with measured impulse re-

[1] `https://dirha.fbk.eu/`

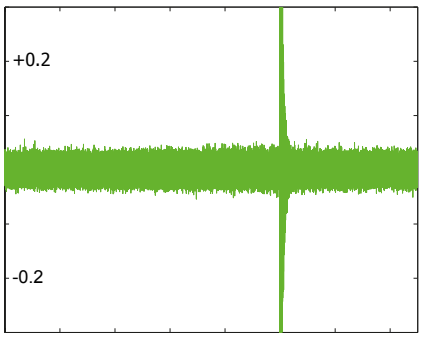
sponses. Our approach will then be validated in Sec. 4.2, reporting a comparison between real and simulated data. Sec. 4.3 studies a novel methodology to derive synthetic IRs. Finally, Sec. 4.4 will describe some approaches to better train DNNs with contaminated data.

4.1 Measuring IRs with ESS

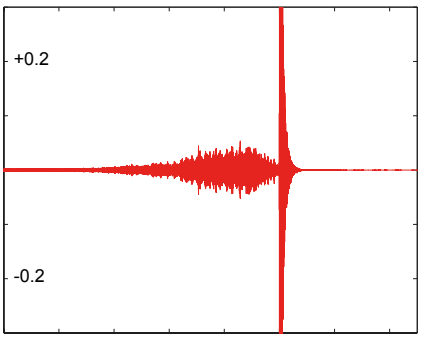
IR estimation in an acoustic enclosure is a topic that has been widely discussed in the literature of the last two decades. The early methods were referred to as *Direct*, i.e. methods based on diffusing in the environment a signal of impulsive nature, as a gun shot or a bursting balloon. These approaches were then replaced by *Indirect* ones, characterized by using excitation signals different from the Dirac function, primarily due to the advantage of providing a higher SNR that was not guaranteed by the former one. In the case of indirect methods, a known excitation signal $s[n]$ of length L is reproduced at a given point (for instance through a loudspeaker), and the corresponding signal $y[n]$ is observed by a microphone placed in another point in space. The related impulse response can be derived by performing a cross-correlation between the known and the recorded signal, as highlighted in the following equation:

$$h[n] = s[n] \star y[n] = \sum_{l=0}^{L-1} s[l] \cdot y[l+n] \tag{4.2}$$

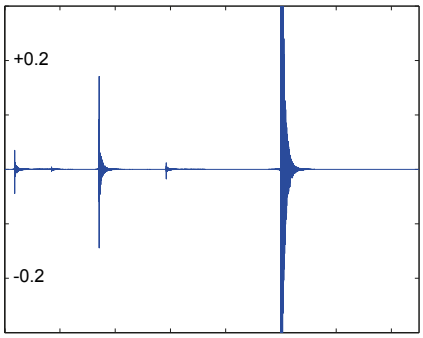
The IR measurement process is affected by environmental noise and non-linearities that may be introduced by instrumentation. In particular, a crucial aspect is the robustness against harmonic distortions, that are non-linear artifacts arising when the loudspeaker does not work in a perfectly linear input-out regime. These artifacts are very common when measuring IRs. In fact, to achieve a high SNR, the monitor should emit a very loud



(a) IR with MLS.



(b) IR with LLS.



(c) IR with ESS.

Figure 4.1: Impulse responses measured in an acoustic enclosure with various indirect methods. To increase the SNR, the loudspeaker is set to a high volume, thus introducing non-linear harmonic distortions in the measurement process.

signal. High volumes, however, typically force the loudspeaker to work in a point of the transfer function closer to saturation, where the input-output relation is not anymore perfectly linear, generating harmonic tones that severely degrade the quality of the measurement. In the category of indirect methods, some of the most commonly used state-of-the-art techniques are *Maximum Length Sequence* (MLS) [249], *Linear Sine Sweep* (LSS) and, more recently, *Exponential Sine Sweep* (ESS) [71], that are signals emitted into the acoustic environment with a duration L and a volume V. The aforementioned methods have a different degree of robustness against noise and harmonic distortions, as described in the following:

- **Maximum Length Sequence (MLS)**. Originally proposed by Schroeder

[249], the MLS technique provides an indirect measurement of IR based on a finite-length pseudo-random sequence of pulses, that has spectral properties almost equivalent to a pure white noise. Based on this technique, the impulse response is derived by a cross-correlation between the MLS sequence and the microphone signal. As mentioned above, if compared to direct methods, MLS offers a better SNR. It is however sensitive to non-linearities introduced by the measurement system, as shown in Figure 4.1a.

- **Linear Sine Sweep (LLS)**. LSS [71] is characterized by an excitation input signal consisting of a sine whose frequency sweeps linearly with time (also referred to as *chirp*). Denoting with ω_1 and ω_2 the initial and final angular frequencies of the sweep, it can be defined as follows:

$$s[n] = \sin\left(\omega_1\, n + \frac{(\omega_2 - \omega_1)}{L}\frac{n^2}{2}\right) \tag{4.3}$$

  As in the case of MLS, the impulse response derives from a cross-correlation between input and output signals. Besides a better SNR than in the MLS case, LSS introduces the advantage of a better (although not perfect) processing of the non-linearities.

- **Exponential Sine Sweep (ESS)**. The ESS technique, introduced by Farina [71], is based on an exponential time-growing frequency sweep, as described by the following relationship:

$$s[n] = sin\left[\frac{\omega_1 \cdot L}{ln\left(\frac{\omega_2}{\omega_1}\right)}\left(e^{\frac{n}{L}\cdot ln\left(\frac{\omega_2}{\omega_1}\right)} - 1\right)\right] \tag{4.4}$$

  An advantage offered by ESS is the immunity against harmonic distortions. As shown by Figure 4.1c, a perfect separation can be observed between the contributions due to harmonic distortions, appearing in

the left part of the estimated IR, and contributions related to the linear impulse response (i.e., reverberation), observable in its right part. Another advantage offered by ESS is that its excitation signal spectrum is pink (note that it's white-like for both MLS and LSS), which ensures to have a better SNR at lower frequencies. This is a desirable feature both at perceptual level, since human auditory system is more discriminative at lower frequencies, and for speech recognition purposes, for which a mel filter-bank is used with higher resolution in the lower part of the frequency axis. Due to this coloration in frequency, ESS impulse responses are characterized by an unnatural dominance of the low-frequency components [71]. A whitening filter with 3dB/octave should thus be applied before computing the cross-correlation to equalize the computed IRs.

In the past, various attempts have been done to compare these impulse response measurement techniques. Our contribution, described in detail in [225], is the comparison of these methods with a specific focus on distant speech recognition. In particular, we compared MLS, LSS and ESS according to some important parameters of the measurement process, including the length L of the emitted signal and its output level V. The training of the DSR system was performed with contaminated versions of the APASCI dataset [5], that is an Italian corpus of phonetically-rich sentences (See App. B for more details about this database). In order to produce an experimental evidence more directly dependent on the acoustic information, the reference task was a word-loop. The test sentences were command and control utterances recorded in the DIRHA apartment by 11 Italian speakers. See App. C.1 for a detailed description of the experimental setup.

A first set of experiments was conducted to investigate on the impact that excitation length L has on recognition performance. In this case, the

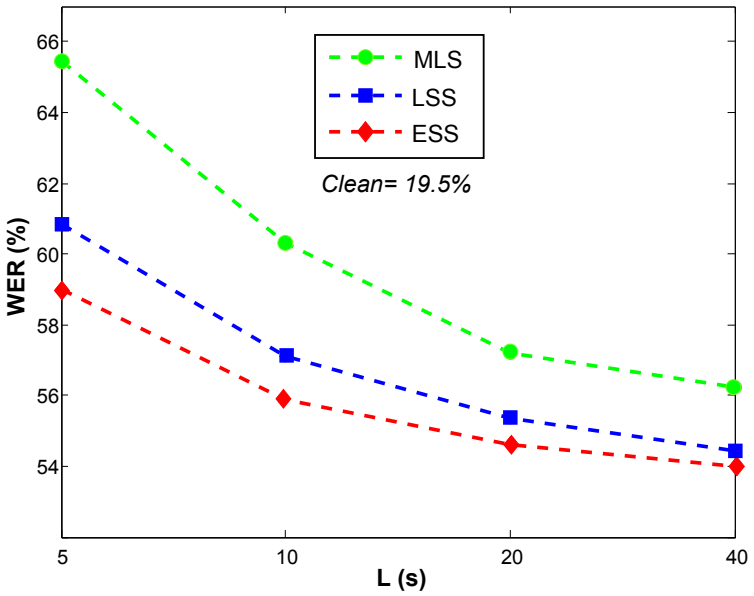


Figure 4.2: Performance in terms of WER (%) obtained with a word loop task when varying the length L of the excitation signal using a professional loudspeaker Genelec 8030.

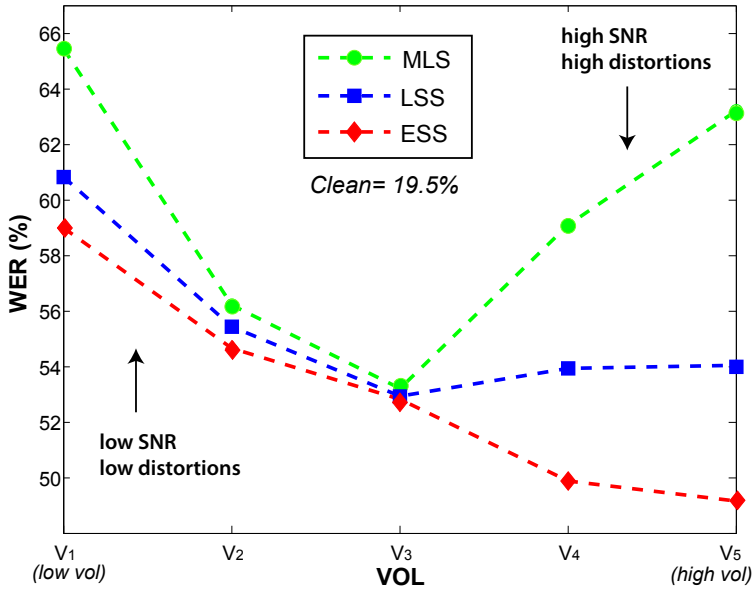


Figure 4.3: Performance obtained with a word loop task when varying the loudspeaker output level using a professional loudspeaker Genelec 8030.

impulse responses were derived based on diffusing in the environment the excitation signals with a low amplitude level. In Figure 4.2, one can note that an increase of L corresponds to an improved performance for all the investigated techniques, thanks to the improved SNR in the IR measurement process. Results also show that ESS provides the best performance at any excitation signal length. This fact can be due to a better SNR at lower frequencies (pink spectrum), e.g. below 2-3 kHz, typically more critical in speech recognition. MLS and LSS, characterized by a white-like spectrum, do not have this interesting property. The experimental results also show that MLS has a higher sensitivity to noise than the other two techniques. However, in general, the difference in performance tends to decrease when L increases.

To study the impact of harmonic distortions, a second set of experiments regarded the analysis of recognition performance when impulse response measurements were realized with different dynamics at loudspeaker output level. From Figure 4.3 it is worth noting that ESS outperforms the other two techniques. As previously observed, for lower dynamics (V_1, V_2), this is due to a better management of SNR. On the other hand, at higher levels of dynamics (V_4, V_5) the best performance provided by ESS is mainly due to a better management of harmonic distortions.

While with MLS and LSS one should choose a trade-off setting in order to have a satisfactory SNR without introducing harmonic distortions, ESS overcomes this trade-off, ensuring a better performance also when the output level of the loudspeaker increases. This study thus clearly demonstrated the superiority of the ESS technique, that was exploited in the following part of this thesis to derive realistic simulated data.

4.2 On the Realism of Simulated Data

Once established a data contamination method, we tried to better validate our approach by performing a more detailed comparison between real and simulated data.

The experiments reported in this section summarize our results obtained in [226, 223], where we tested the realism of contaminated data under a variety of experimental conditions. The reference scenario was the DIRHA apartment, that was equipped with the microphone setup depicted in Figure 4.4 and described in detail in App. A.2.

The experiments involved five US speakers that recorded a set of sentences extracted from the wall street journal. To acquire high-quality close-talking material, a first speech acquisition was performed in our recording studio. Using the approach described in the previous section, we then measured several impulse responses in the living-room of the DIRHA apartment and we generated a simulated dataset. To record real data well-matching with this simulated corpus, we asked the same speakers to utter the same sentences in the same positions of the DIRHA living-room used for measuring the impulse responses. Thanks to this alignment, we were able to perform a comparison as fair as possible between real and simulated data. See the Appendix for more detail on the experimental setup (App. C.2).

The results reported in the first column of Table 4.1 show the performance obtained when a single distant microphone (i.e., the "$LA6$" ceiling microphone depicted in Figure 4.4) is considered. The "*mono*" model is a context-independent GMM model (monophones), "$tri4$" is a context-dependent GMM model based on Speaker Adaptive Training (SAT), while "DNN" represents the considered FF-DNN model.

Results clearly highlight that in the case of distant-speech input the ASR performance is dramatically reduced, if compared to a close-talking case

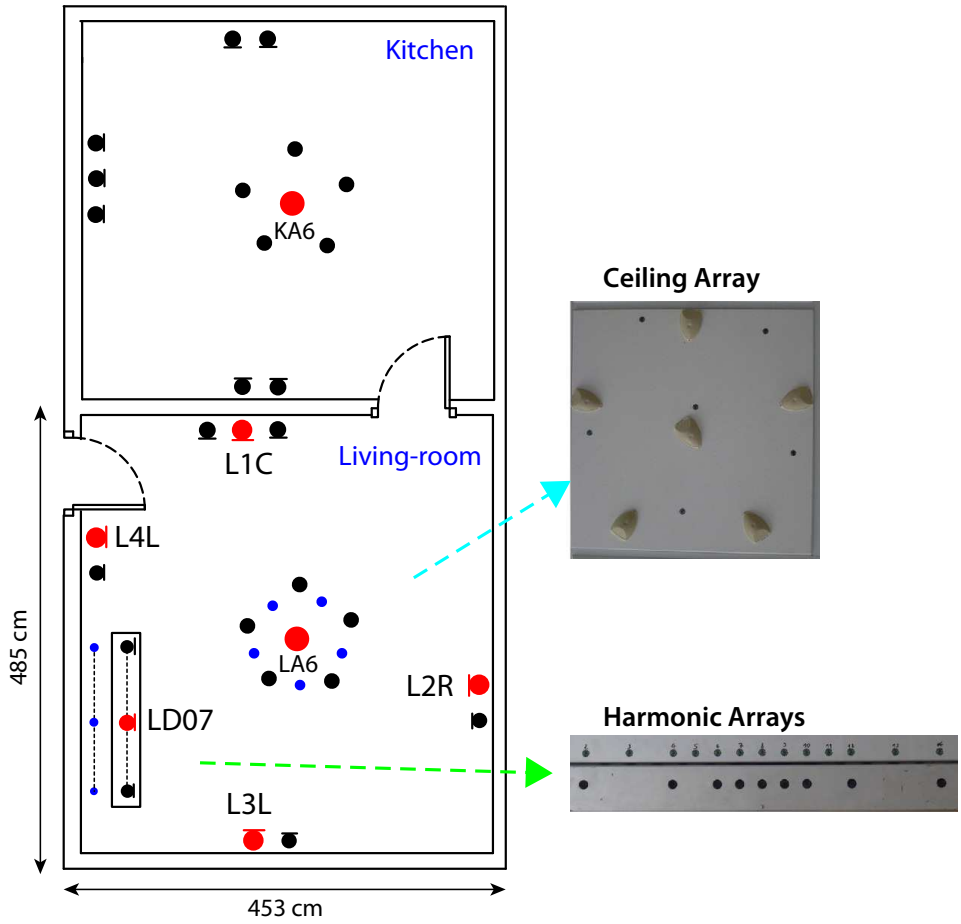


Figure 4.4: An outline of the microphone set-up adopted for the real recordings and for IR measurement. Blue small dots represent digital MEMS microphones, red ones refers to the channels used for most of the experiments, while black ones represent the other available microphones. The right pictures show the ceiling array and the two linear harmonic arrays installed in the living-room.

(WER=3.7%). The use of robust DNN models trained with contaminated speech material leads, as expected, to a substantial improvement of the WER when compared to other GMM-based systems[2]. The most interesting result, however, is that a similar performance trend is obtained for both real and simulated data over different acoustic models. This trend can

[2]Note that GMM-HMMs systems, whose performance is not anymore competitive with modern DNN-based systems, are reported here only to analyze the similarity between real and simulated data using different acoustic models.

	Single Distant Microphone		*D&S Beamforming*	
	Real Data	Sim Data	Real Data	Sim Data
Mono	62.2	64.7	56.8	58.8
Tri4	19.9	21.4	17.5	17.4
DNN	12.0	13.2	10.7	11.6

Table 4.1: WER(%) obtained in a distant-talking scenario with real and simulated data across different acoustic models and microphone processing.

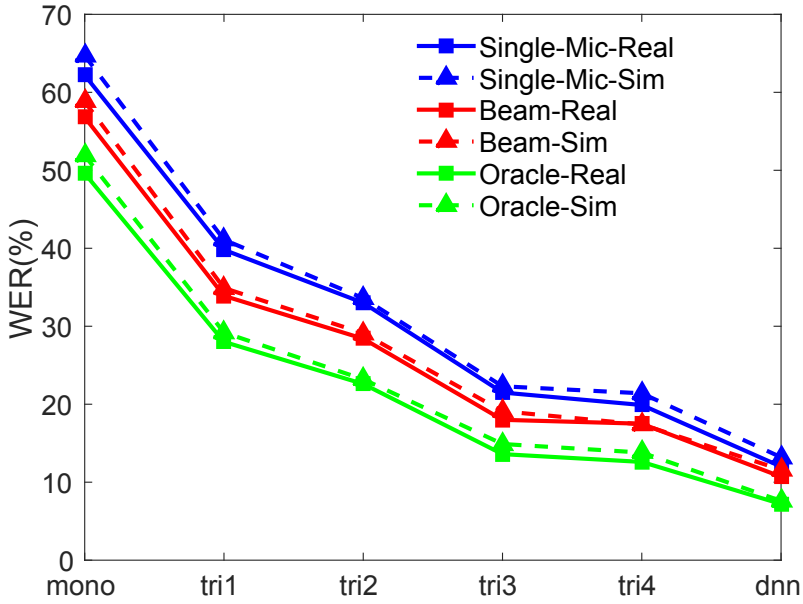


Figure 4.5: Comparison of the performance trends obtained with real and simulated data under a variety of experimental conditions.

also be appreciated by comparing the continuous (real data) and dashed (sim data) blue lines of Figure 4.5. The average relative WER distance between such data-sets computed over the considered acoustic models is about 6%. We believe that this is a significant result, especially if one takes into account that part of this variability can be attributed, despite our best efforts for aligning simulated and real data, to the fact that in the two recording sessions (i.e, the close-taking recordings to derive simulated data and the real distant speech acquisitions in the apartment) speakers inevitably uttered the same sentence in a different way.

The simulation methodology described in Sec. 4.1 can be extended in a very straightforward way to a multi-microphone scenario. It would be thus of crucial importance to ensure that the similar trend between real and simulated data achieved with a single microphone is preserved even when multi-microphone processing is applied to the data. In the following experiment, we considered the six microphone array placed on the ceiling of the livingroom (see Figure 4.4) and applied a delay-and-sum beamforming to both real and simulated data. As described in Sec. 3.3.1, the source-microphone delays have been computed with the GCC-PHAT algorithm [192].

Table 4.1 and Figure 4.5 show that beamforming is helpful for improving the system performance (see the red lines). One can also note that, as hoped, a similar performance trend between the datasets is reached when applying beamforming. For instance, in the case of real data coupled with FF-DNN acoustic models, delay-and-sum beamforming leads to a relative improvement of about 12% over the single microphone case, which is similar to the improvement of 13% obtained with the simulated data.

Another way to compare real and simulated data in a multi-microphone scenario is to perform a microphone selection. In the following experiment, we considered the six microphones of the DIRHA livingroom depicted as red circles in Figure 4.4. In particular, for each sentence uttered by the speaker, the best WER from the signals acquired by the microphones was considered. This experiment, called *oracle microphone selection*, is reported here only to provide an upper bound of the DSR performance, since real microphone selection techniques are still far from this ideal case [288, 105]. We anyway believe that this test is particularly interesting because the results might depend on the directional characteristics of the speakers in real and simulated data. Simulated data, in fact, inherit the directivity of the loudspeaker used to measure the impulse responses, and it

would be of interest to understand if this aspect affects the spatial realism of our simulations.

Figure 4.5 shows the WER achieved with the oracle microphone selection (see green lines). Results confirm that the consistency between real and simulated data is largely preserved. This would suggest that simulated data are able to represent the directional/spatial properties of the speaker with a sufficient level of realism. The experimental results also show that an optimal microphone selection would be particularly helpful to improve the DSR performance.

Thanks to the remarkable level of realism obtained with the proposed approach, several multi-microphone datasets have been developed in the context of the DIRHA project and publicly distributed at international level. The list of corpora based on our realistic data contamination approach is the following:

- DIRHA Simcorpora [51]
- DIRHA-English [216]
- DIRHA-GRID [167]
- DIRHA-AEC [306]

See the Appendix (App. B) for a detailed description of each corpus.

4.3 Directional Image Method

When directly measuring an impulse response is not possible, an alternative is to synthetically derive it with a room simulator. Concerning this, different methods are described in the literature. The image method (IM) [3], proposed by Allen and Berkley in 1979, is the most commonly used technique in the speech recognition community. IM refers to the so-called wave

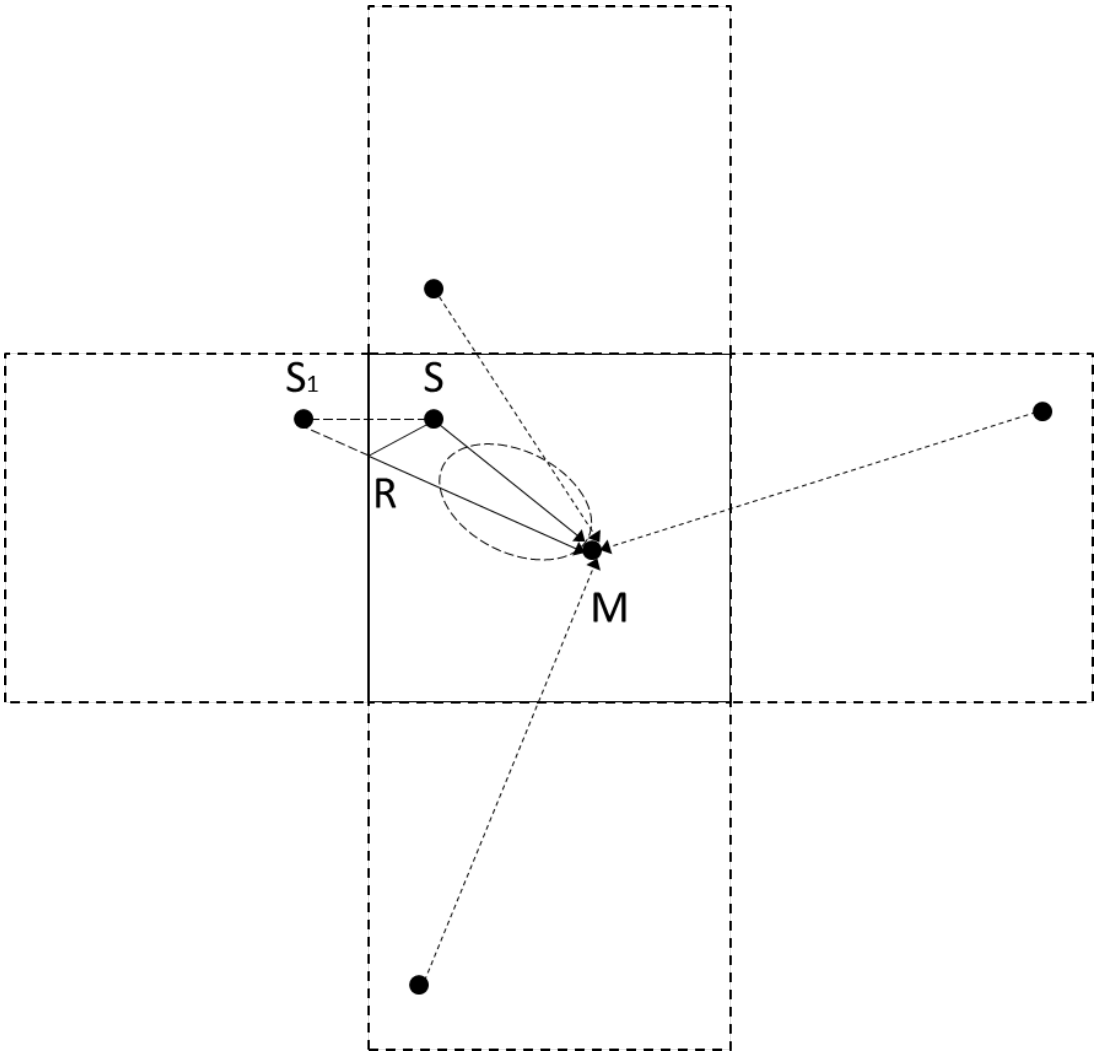


Figure 4.6: Basic principle behind the implementation of the image method.

equation, and to its frequency domain counterpart called the Helmholtz equation, which describes wave propagation in a fluid [147].

Fig. 4.6 shows the image model used for the 2-D implementation of the standard image method, where a microphone M and a source S are considered. The source, that in the original formulation proposed in [3] is omidirectional, radiates the acoustic signal in all the directions. To compute the impulse response, the delay τ of each possible path connecting the microphone with the source should be computed. These delays depend

on the length l of each path and on the sound velocity c:

$$\tau = \frac{l}{c} \tag{4.5}$$

The length of the direct path can be computed from the known positions of the source and the microphone. To compute the length of the reflection paths, the IM method exploits an efficient procedure based on the concept of images. For instance, a first order reflection that follows the path $SR-RM$ is depicted in Fig. 4.6. This reflection can be considered as generated by an equivalent image source S_1, that is obtained by mirroring the original source S over the reflection wall. The triangle SS_1R is isosceles and therefore the path length $SR + RM$ is the same as S_1M. This means that the length of the reflection path corresponds to the distance between the image and the reference microphone. The other first order reflections can be derived by mirroring the source over the different walls, while higher order reflections can be computed by progressively mirroring the previous images. The attenuation α of each reflection $\delta(t-\tau)$ depends on the path length l and on the number n of reflections involved in the path:

$$\alpha = \frac{\rho^n}{4\pi l} \tag{4.6}$$

where ρ is the reflection coefficient (ranging from 0 to 1) that described the reflection properties of the environment.

Different algorithms based on IM are described in the literature and some software tools have been made available to the scientific community [154, 107]. In the last decades, several modifications of original Allen-Berkley's algorithm have been proposed. Some of them consider the extension to 3D room acoustics, a different reflection coefficient for each surface, an implementation in the frequency domain, and the simulation of the microphone polar pattern (e.g., omnidirectional, or cardioid), as described in the literature referenced above.

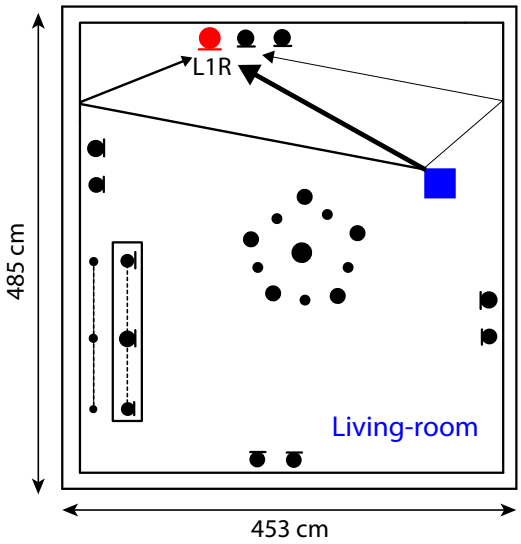
In particular, the simulation of directional polar patterns is rather straightforward within the image method framework. As shown in Fig. 4.6, the direction of arrival of the reflection (RM) is the same of the image-microphone path S_1M. To account directivity it is thus sufficient to multiply the strength of the reflection by a gain factor $D(\theta)$, that depends on the related angle between microphone orientation and reflection.

Beside microphone directivity, we believe that another fundamental aspect concerns the simulation of sound source directivity, that contributes substantially in characterizing the process of speech propagation in space. As reported in [181], speech directionality depends not only on the speaker's head pose but also on other factors, as speaker's gender, mouth shape and consequently uttered phonemes. As a matter of fact, the use of an impulse response that derives from an omnidirectional source hypothesis is not sufficient to obtain the required realism, if compared with real speech acquired under equivalent conditions.

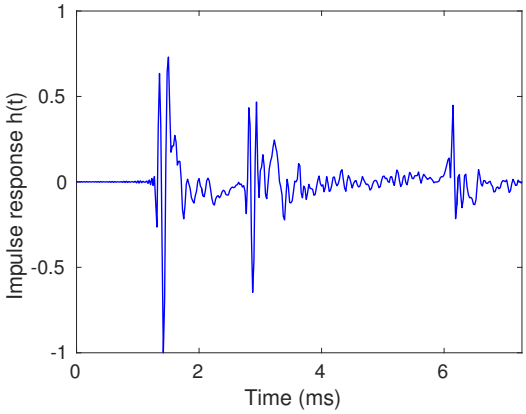
The importance of a proper directivity matching between real and simulated data has been reiterated in the previous section. Differently to previous experiments, that were based on measured impulse responses, in this section we elaborate on directivity for synthetic IRs. In this work, we used a software available at FBK, which simulates sound source directivity, based on a modification of the original IM algorithm. This modification simply exploits the principle of reciprocity: instead of mirroring the source, the microphone is mirrored. The various paths connecting the images are then multiplied not only by α but also by the following directivity factor:

$$D(\theta) = \frac{\left(\frac{1+cos(\theta)}{2}\right)^p + \epsilon}{1+\epsilon} \tag{4.7}$$

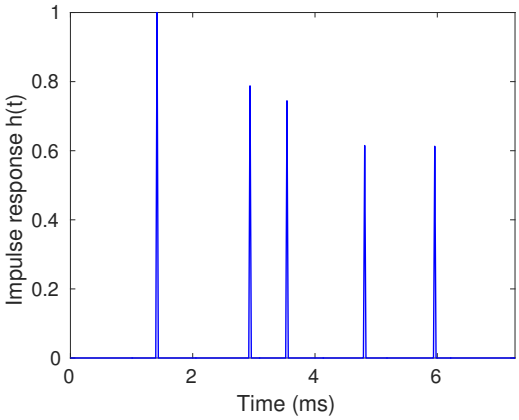
where θ is the relative angle between source orientation and direction of the reflection, p is a parameter used to tune the source direcitivity factor,



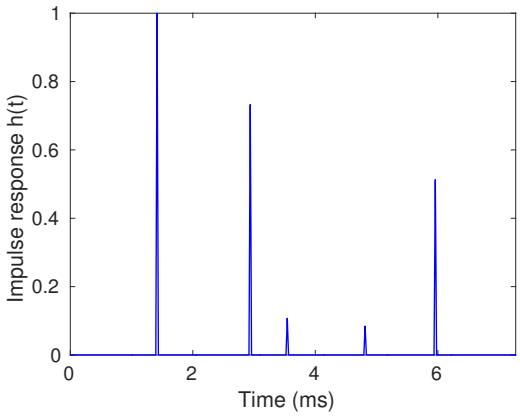
(a) Room Geometry.



(b) Measured impulse response.



(c) Original Omnidirectional IM.



(d) Directional IM.

Figure 4.7: Early reflections obtained from impulse responses measured/simulated in the DIRHA livingroom with the room geometry depicted in (a). The thickness of the highlighted paths is proportional to the energy diffused along that angle by a directional sound source.

and ϵ is a small constant to avoid paths with zero gain. Note that there are actually both azimuth D_{az} and elevation D_{el} directivities in the considered 3-D implementation of the image method. The total directivity can be thus obtained as follows:

$$D_{az}(\theta) = \Big(\frac{1 + cos(\theta)}{2}\Big)^{p} \tag{4.8}$$

$$D_{el}(\phi) = \Big(\frac{1 + cos(\phi)}{2}\Big)^{q} \tag{4.9}$$

$$D_{tot}(\theta, \phi) = \frac{D_{az}(\theta) \cdot D_{el}(\phi) + \epsilon}{1 + \epsilon} \tag{4.10}$$

where p and q are the azimuth and elevation directivity factors, respectively.

The experiments reported in the remaining part of this section are based on the experimental evidence emerged in [223]. Values of $p = 3$, $q = 1$, and $\epsilon = 0.01$ have been used for the following experiments.

A qualitative comparison between the considered directional IM and a more standard implementation of the IM is reported in Figure 4.7. Figure 4.7b shows the early reflections of an impulse response measured in the DIRHA apartment, while Figure 4.7c shows a corresponding IR simulated with the standard omnidirectional image method. Finally, Figure 4.7d represents the directional implementation. Note that the source-microphone position and orientation depicted in Figure 4.7a were considered for all the reported IRs.

The first peak, that is clearly highlighted in all the IRs, is the direct path. The other ones correspond to reflections on the walls, floor and ceiling of the living-room. It is worth noting that the proposed method originates reflections with an amplitude more similar to that encountered in the real case (compare Figure 4.7b and Figure 4.7d). In the omnidirectional case, for instance, the third and fourth reflections have a significant

IR ID	IM-OMNI	IM-DIR	IR-MEAS
IR type	*IM-method*	*IM-method*	*Measured*
IR directivity	*omni*	*directional*	*directional*
WER(%)	13.0	12.5	12.0

Table 4.2: WER(%) obtained on real data by contaminating the training data with different implementations of the image method algorithm.

magnitude. On the other hand, the amplitude of those peaks is close to zero in both the measured and directional IR, thanks to the assumption of source directionality. Properly managing this aspect, thus significantly contributes to generate synthetic IRs that are more similar to the corresponding measured ones. As a matter of fact, it is also worth noting that a perceptually more natural simulated speech signal is provided by contaminating clean speech with the latter IR.

In the following experiment, we extended our study to the speech recognition domain. Training was based on contaminated versions of the WSJ dataset, using impulse responses estimated in the reference DIRHA living-room. Test was based on the real recordings described in the previous section. See App. C.2 for more details.

Table 4.2 shows the recognition results obtained with speech material contaminated with different IRs deriving from the considered implementations of the image method. As expected, the best performance is achieved with the measured IRs (IR-MEAS). It is interesting to note, however, that the considered directional implementation (IM-DIR) outperforms the omnidirectional version of the image method (13.0% vs 12.5%).

Our results confirmed the importance of simulating source directivity within the IM framework. Taking into account sound source directionality while training the acoustic model indeed implies a better match with real data, that improves the performance of the speech recognizer. We would like to highlight some analogies with what observed in [32], where

ASR performance highly correlated with early-to-late reverberation ratio. In general, taking into account sound source directionality while training the acoustic model implies that a better match with real data will be obtained as far as DRR (and, consequently, other early-to-late reverberation features) is concerned. A correlation between DRR and recognition performance has also been evidenced in another recent work from us [76]. In general, DRR depends on the distance between source and microphone, and it is much more negative when an omnidirectional sound source is simulated, if compared to the case of a directional sound source.

4.4 DNN Training with Contaminated Data

After proposing a methodology for realistic data contamination, we focused on the study of proper approaches to train DNNs with contaminated data. In particular, we tried to exploit an interesting peculiarity of contaminated datasets, i.e. the fact that the same sentences are available in both a close- and distant-talking conditions.

The following sections summarize the experiments reported in [220]. The experimental validation was based on a phone-loop task, in order to achieve an experimental evidence not biased by language information. The reference environment was the DIRHA livingroom, that was equipped with the microphone setup described in the Appendix (see App. A.1). Training was based on contaminated versions of both APASCI [5] and Euronews [101] datasets, while test was performed on real and simulated sentences of the DIRHA-phrich dataset. In this work, we considered acoustic conditions of increasing complexity by adopting simulated data with reverberation (Sim-Rev) and with both noise and reverberation (Sim-$Rev\&Noise$). For more details on the experimental setup see the Appendix (App. C.3).

Close-Talking Labels

In standard ASR, the labels for DNN training are derived by a forced-alignment of the training corpus over the tied-states. Although some GMM-free solutions have been proposed [254], this alignment is typically performed using a standard CD-GMM-HMM system. This phase can be very critical because a precise alignment could be difficult to reach, especially in challenging acoustic scenarios characterized by noise and reverberation. As a consequence, the DNN learning process might be more problematic due to a poor supervision.

In the contaminated speech training framework, however, a more precise supervision can be obtained from the close-talking dataset. Although this is a natural choice for this kind of training modality, our contribution is to better quantify the benefits deriving from this approach. The studied methodology requires to train a standard CD-GMM-HMM system with the original clean datasets, and exploit it to generate a precise tied-state forced alignment over the close-talking training corpus, later inherited as supervision for the distant-talking DNN.

Table 4.3 reports the results obtained with the standard approach, that is based on labels derived from distant-talking signals, and the proposed close-talking variation. Results show that the proposed approach provides a substantial performance improvement, over both real and simulated data. Specifically, a relative improvement of 10% and 12% was achieved for APASCI and Euronews, respectively. Results shows that the benefits of this approach holds also under challenging acoustic conditions characterized by noise and reverberation.

Besides this significant reduction of the error rates, another interesting aspect is the faster convergence of the learning procedure, due to a better supervision provided to the DNN. In these experiments, 15 epochs were

Train / *Test*	APASCI (6 h)		Euronews (100 h)	
	Standard	CT-lab	Standard	CT-lab
Sim-Rev	37.0	**33.0**	36.1	**32.1**
Real-Rev	40.2	**35.9**	39.3	**34.3**
Sim-Rev&Noise	51.8	**47.3**	50.1	**46.4**

Table 4.3: PER(%) obtained in distant-talking scenarios with the standard and with the proposed technique based on close-talking labels (CT-lab).

Train / *Test*	APASCI (6 h)		Euronews (100 h)	
	Stand-PT	CT-PT	Stand-PT	CT-PT
Sim-Rev	33.0	**31.4**	32.1	**31.2**
Real-Rev	35.9	**34.7**	34.3	**33.0**
Sim-Rev&Noise	47.3	**46.2**	46.4	**45.1**

Table 4.4: PER(%) obtained in distant-talking scenarios with a standard RBM pre-training (Stand-PT) and with the proposed supervised close-talking pre-training (CT-PT).

needed to converge with the standard solution, while only 12 were sufficient with the proposed approach, reducing the training time of 20%.

Supervised Close-talking Pre-Training

A proper DNN initialization is of crucial importance in the context of deep learning, as outlined in Chapter 2. One of the first methodologies proposed in the literature consisted to initialize DNNs using unsupervised data. The most popular approach was pre-training based on Restricted Boltzman Machines (RBM), that turned out to discover higher level feature representations starting from low-level ones.

In this work, we studied a supervised alternative to RBM pretraining, which was able to take advantage of the rich information embedded in the close-talking dataset. In particular, we proposed to use a supervised pre-training method based on the close-talking data. The idea is to

train a close-talking DNN and inherit its parameters for initializing the distant-talking DNN. A subsequent fine-tuning phase is then carried out on distant-talking data using a slightly reduced learning rate. This idea resembles a curriculum learning strategy [18], since we actually propose to first solve a simpler close-talking problem and address the challenging task of environmental robustness only as a second step.

Results, reported in Table 4.4, clearly show a performance improvement, that is consistent for both APASCI and Euronews datasets. This improvement is obtained over both real and simulated data and also arises in the challenging scenarios characterized by both noise and reverberation (*rev&noise*).

This suggests that a close-talk pre-training is a smart way of initializing the DNN, which somehow first learns the speech characteristics and only at a later stage learns how to counteract adverse acoustic conditions.

4.5 Summary and Future Work

This Chapter summarized our efforts on speech contamination. First of all we studied a methodology for measuring high-quality impulse responses with the ESS method. We then validated our contamination approach with an extensive experimental activity carried out on both real and simulated data. Finally, we proposed some original methodologies for training DNNs with our simulated data.

This Chapter has also preliminarily addressed the problem of deriving realistic synthetic impulse responses. We proposed a modified version of the original image method that considers directional acoustic sources. This algorithm is being extended to a frequency domain implementation, that promises to generate more realistic impulse responses. Despite our best efforts, current synthetic impulse responses are still far from accurately

modeling the actual complexity of a real acoustic environment. For example, typical acoustic enclosures are composed of multiple scattering objects, that can significantly alter the overall impulse response. Even though it is currently not possible to directly address such a level of complexity, we believe that processing synthetic impulse responses with some proper noisy sequences might help to better mimic the presence of multiple object spread in the acoustic enclosure. Some future studies will be focused on this topic.

Chapter 5

Exploiting Time Contexts

In the previous Chapter, we emphasized the importance of contaminated data for improving the performance of a speech recognizer. Another pivotal aspect to increase DSR robustness is a proper analysis of time contexts. Speech, indeed, is inherently a sequential non-stationary signal that evolves dynamically, making of paramount importance the study of methodologies to manage temporal dynamics. This need is particularly important for distant speech recognition, where contexts might help counteract the considerable uncertainty introduced by noise and reverberation.

Actually, the importance of modeling speech contexts is known since long time. In the case of previous HMM-GMM systems, some attempts were done for better exploiting short-term speech dynamics. For instance, high-order derivatives and LDA/HLDA transformations of concatenated frames represented very popular techniques to broaden time contexts [121]. However, HMM-GMM systems were largely inadequate for managing longer-term information, due to both their inability to handle high-dimensional input spaces and their limitations in dealing with correlated feature vectors. A key reason behind the current success of deep learning is the natural ability of DNNs to better manage large time contexts [196].

Despite the progress of the last years, we believe that it is of great

interest to continue the study of techniques for better processing long-term information.

In this thesis we contributed to this area of research considering both feed-forward and recurrent neural networks. The remaining part of this Chapter summarizes the main results reported in the papers we published on this topic [221, 222, 217, 218, 213, 214]. To better contextualize our contribution, in Sec. 5.1 we first propose a brief introduction of the main techniques adopted to manage short-term information with feed-forward DNNs. Sec. 5.2 then reports our study on an asymmetric context window [221, 222]. In Sec. 5.3 we extend our discussion to the most popular methodologies for embedding long-term information in DNN acoustic models. Finally, Sec 5.4 summarizes our recent efforts to revise standard Gated Recurrent Units (GRU) [213, 214].

5.1 Analysing Short-Term Contexts

The most straightforward way to exploit time contexts for feed-forward neural networks is to directly feed the DNN with multiple frames. This standard approach is depicted in Figure 5.1, where the distant-talking signal $y[n]$ captured by the far microphone is processed by a feature extraction function $f(\cdot)$ that computes a sequence of features frames. For each frame $\mathbf{y_k}$, a DNN is employed to perform frame-level phone predictions. In order to estimate more robust posteriors, the network is not only fed with the current frame $\mathbf{y_k}$, but also with some surrounding ones. The set of frames feeding the DNN is often called *context window*. Indicating with $\mathbf{y_k}$ the distant-talking feature vector for the k-th frame of a speech sentence, the context window is the set of frames defined as follows:

$$CW_k = \{\mathbf{y_{k-N_p}}, ..., \mathbf{y_k}, ..., \mathbf{y_{k+N_f}}\} \tag{5.1}$$

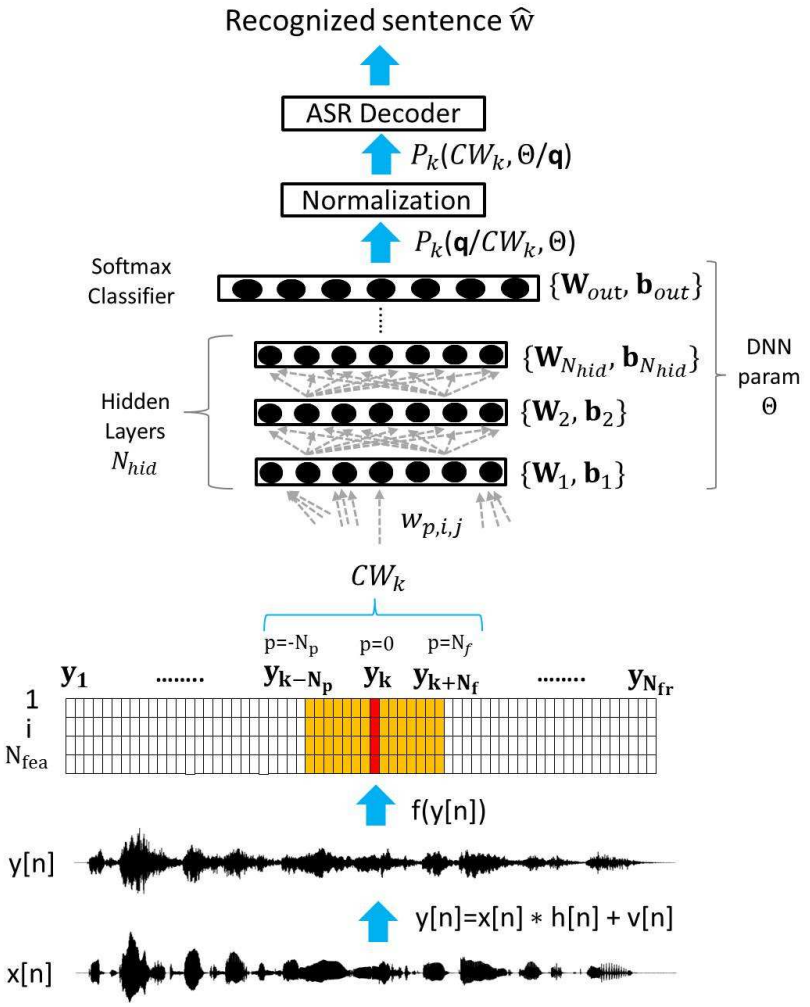


Figure 5.1: End-to-end pipeline of a HMM-DNN distant speech recognizer. The system is fed with a set features derived from the distant-talking signal $y[n]$.

where N_p and N_f are the number of past and future frames. Although this simple approach works relatively well in most of the cases, the input dimensionality can be too high when concatenating several frames, possibly causing curse of dimensionality issues. To mitigate such a concern, LDA [211] or DCT [274] can be used to perform a dimensionality reduction.

In this thesis we contributed to study a novel context window configuration, that turned out to be suitable for counteracting environmental reverberation. The proposed context window setting, called asymmetric

context window, have been described in [221, 222], and is summarized in the following section.

5.2 Asymmetric Context Window

The vast majority of past works on feed-forward DNNs consider symmetric context windows for frame concatenations (i.e., $N_p = N_f$). Actually, only very few papers adopted asymmetric windows ($N_p \neq N_f$) in their experimental settings, mainly for real-time applications in close-talking scenarios [155, 43, 200]. Differently to previous works, we found this approach not only appropriate for low-latency systems, but also better performing than traditional window mechanisms in distant-talking conditions. In particular, asymmetric contexts that embeds more past than future frames ($N_p > N_f$), turned out to be effective to tackle reverberation.

To account for different balance factors between past and future frames, a useful coefficient ρ_{cw} that will be used in this work is defined as follows:

$$\rho_{cw}(\%) = \frac{N_p}{N_p + N_f} \cdot 100 \tag{5.2}$$

The asymmetric windows proposed in this work are based on $\rho_{cw} > 50\%$. Under reverberant conditions, the proposed windowing mechanism has proven to be a viable alternative to a more standard symmetric context. The asymmetric window, in fact, feeds the DNN with a more convenient frame configuration which carries, on average, information that is less redundant and less affected by the correlation effects introduced by reverberation.

Although our work was not the first one proposing asymmetric context windows, our contribution is, to the best of our knowledge, the first attempt to extensively study the role played by the asymmetric context window to counteract reverberation.

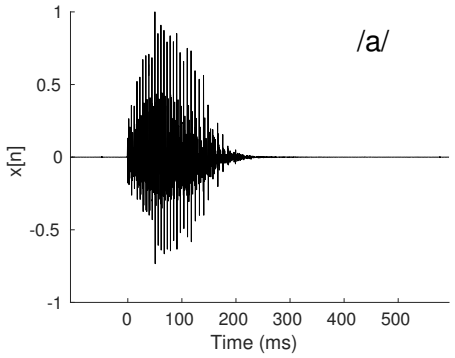
The experimental evidence reported in the following summarizes the main results obtained in [221, 222]. In particular, a detailed validation has been carried out to show the effectiveness of the investigated method. The following section first provides evidence at signal and feature levels by performing a correlation analysis. In Sec. 5.2.2 we then analyse the DNN weights and, finally, Sec. 5.2.3 reports the speech recognition experiments.

5.2.1 Correlation Analysis

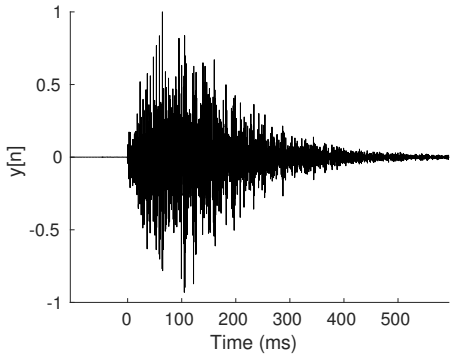
A useful tool to study the redundancy in a reverberated speech signal is the cross-correlation. For instance, we can study the effect of reverberation on a speech signal by analyzing the cross-correlation R_{xy} between the close-talking speech $x[n]$ and the corresponding distant-talking sequence $y[n]$. With this purpose, we can expand the signal $y[n]$, defined in Eq. 3.1 of Sec. 3.1, in this way:

$$y[n] = \sum_{m=0}^{M-1} x[n-m] \cdot h[m] \tag{5.3}$$

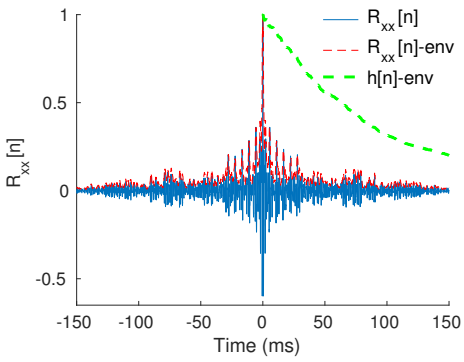
Note that, in order to focus our analysis on reverberation effects only, the additive noise $n[n]$ is omitted here. If we assume $x[n]$ to be a sequence of length L, and $h[n]$ to be an impulse response of length M, the cross-correlation R_{xy} is defined as follows:



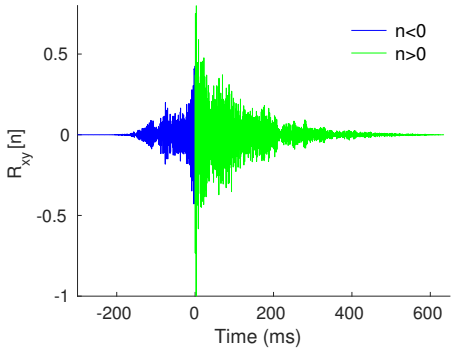
(a) Close-talking signal $x[n]$



(b) Reverberated signal $y[n]$



(c) Autocorrelation $R_{xx}[n]$



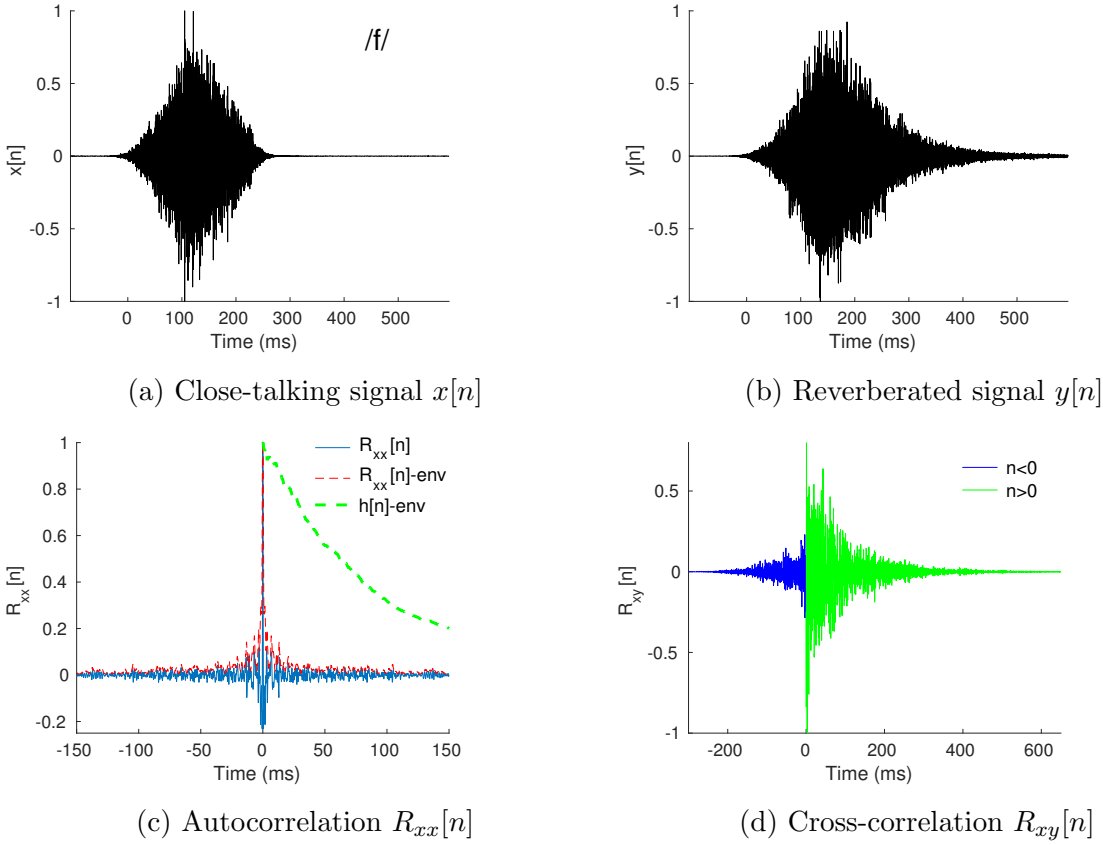
(d) Cross-correlation $R_{xy}[n]$

Figure 5.2: Cross and autocorrelation correlation analysis for the vowel /a/.

$$
\begin{aligned}
R_{xy}[n] &= \sum_{l=0}^{L-1} x[l] \cdot y[l+n] \qquad (5.4)\\
&= \sum_{l=0}^{L-1} x[l] \cdot \left(\sum_{m=0}^{M-1} x[l+n-m] \cdot h[m] \right)\\
&= \sum_{l=0}^{L-1} \sum_{m=0}^{M-1} x[l] \cdot x[l+n-m] \cdot h[m]\\
&= \sum_{m=0}^{M-1} h[m] \underbrace{\sum_{l=0}^{L-1} x[l] \cdot x[l+n-m]}_{R_{xx}[n-m]}\\
&= \sum_{m=0}^{M-1} h[m] \cdot R_{xx}[n-m]
\end{aligned}
$$

The cross-correlation $R_{xy}[n]$ thus depends on both the impulse response $h[n]$ and the autocorrelation function $R_{xx}[n]$. The autocorrelation $R_{xx}[n]$ varies significantly according to the particular phoneme and the signal characteristics that are considered. Figure 5.2c, for instance, shows the autocorrelation $R_{xx}[n]$ of a vowel $/a/$, while Figure 5.3c illustrates $R_{xx}[n]$ for a fricative $/f/$.

One can easily observe that different autocorrelation patterns are obtained: for the vowel sound $/a/$, $R_{xx}[n]$ is based on several peaks due to pitch and formants, while for $/f/$ a more impulse-like pattern is observed. The spread of the autocorrelation function around its center $t = 0$ also depends on the specific phoneme. If we consider, for instance, the time instant where the energy of $R_{xx}[n]$ decays to 99.9% of its initial value, the autocorrelation length is 104 ms with the vowel /a/, and about 25 ms with the fricative /f/. In both cases, however, the duration of the autocorrelation is significantly smaller than the length of the impulse response (see green dashed line of Figure 5.2c and Figure 5.3c). This characteristic,



(a) Close-talking signal $x[n]$

(b) Reverberated signal $y[n]$

(c) Autocorrelation $R_{xx}[n]$

(d) Cross-correlation $R_{xy}[n]$

Figure 5.3: Cross and autocorrelation correlation analysis for the fricative /f/.

together with the causality of the impulse response ($h[n] = 0 \quad \forall n < 0$), originates an asymmetric trend in the cross-correlation $R_{xy}[n]$, which can be clearly appreciated from both Figure 5.2d and Figure 5.3d. As shown by the latter examples, with medium-high T_{60} the right side of this function is influenced by the impulse response decay. The future samples ($n > 0$) are thus, on average, more redundant than previous ones ($n < 0$), and this effect is amplified when reverberation increases.

Based on these observations, we began to realize the possible benefits arising with an asymmetric context window that integrates more past than future frames. This solution, in fact, would be more appropriate than a traditional symmetric context since it results in a frame configuration less affected by the aforementioned forward correlation effects of reverberation.

In other words, with the asymmetric context we can feed the DSR system with information which is, on average, more complementary than that considered in a standard symmetric frame configuration, allowing the DNN to perform more robust predictions.

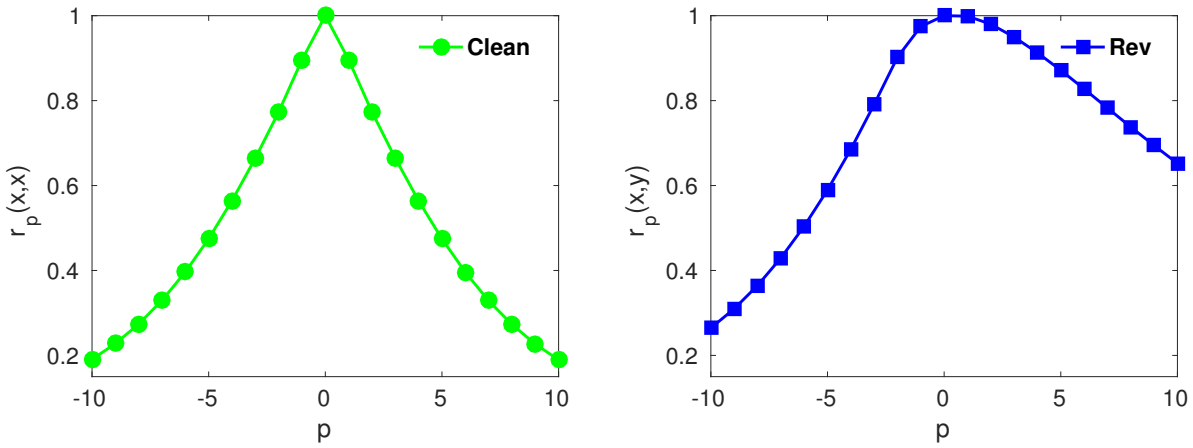
To further validate this conjecture, we extended the previous signal-based results to the feature domain, using the Pearson correlation [199, 13]. This coefficient, denoted as $r_{x,y}(p)$, is computed between close-talking $\mathbf{x}$ and distant-talking $\mathbf{y}$ feature sequences, and it is defined as follows:

$$r_{x,y}(p) = \frac{\sum_{k=1}^{N_{fr}} (\mathbf{x_k} - \bar{\mathbf{x}}) \cdot (\mathbf{y_{k+p}} - \bar{\mathbf{y}})}{\sqrt{\sum_{k=1}^{N_{fr}} (\mathbf{x_k} - \bar{\mathbf{x}})^2} \cdot \sqrt{\sum_{k=1}^{N_{fr}} (\mathbf{y_{k+p}} - \bar{\mathbf{y}})^2}} \tag{5.5}$$

where $\bar{\mathbf{x}} = \sum_{k=1}^{N_{fr}} \mathbf{x_k}$ and $\bar{\mathbf{y}} = \sum_{k=1}^{N_{fr}} \mathbf{y_k}$.

The correlation $r_{x,y}(p)$, is here estimated for $-N_p \leq p \leq N_f$ integrating over all the frames N_{fr}. The argument p denotes the frame lag between the input sequences. For instance, when $p = 0$ the Pearson coefficient $r_{x,y}(0)$ is computed by considering the k-th frames for both close- and distant-talking features. When $p = m$, the correlation is computed between the k-th close-talking frames and $k + m$ frame of the distant talking feature sequence.

We computed the Pearson correlation using the DIRHA-English-WSJ5k dataset (simulated part of the set1 portion, see App. B), considering 13 static MFCC features for both the close-talking recordings and the corresponding reverberated simulations. Note that simulations were generated by compensating time delays with the close-talking recordings. The Pearson coefficient r_p, is reported in Figure 5.4 averaging the correlation for all the sentences of this dataset. In particular, Figure 5.4a reports the auto-correlation $r_{x,x}(p)$ obtained for the close-talking case. From the fig-



(a) Close-talking scenario (Clean) (b) Distant-talking scenario (Rev)

Figure 5.4: Pearson Correlation r_p across the various frames composing the context windows.

ure it is clear that in the close-talking case a symmetric trend is obtained, showing that past and future frames are equally correlated. Figure 5.4b reports the Pearson cross-correlation coefficients $r_{x,y}(p)$ between close- and distant-talking signals.

In this case, an asymmetric correlation trend is observed, highlighting that future frames ($p > 0$) are consistently more correlated than past ones ($p < 0$).

Similarly to what observed for the signal cross-correlation $R_{xy}[n]$, the correlation unbalance of the Pearson coefficients further confirms the potential of the proposed asymmetric context window.

5.2.2 DNN Weight Analysis

Another interesting way to analyze the effect of reverberation is to inspect the DNN weights. In particular, we are interested in studying whether the network is able to automatically assign different importance to the different frames composing the context window. With this regard, it is helpful to analyze the weights connecting the input frames with the first-layer

neurons. If one frame is considered important by the network, we expect that, on average, the corresponding weights have a higher magnitude. On the other hand, if the network assigns little importance to a certain input, its weight connections should have a magnitude close to zero. This directly stems from the role played by the gradient in the context of the SGD optimization. For a weight $w_{p,i,j}$ connecting the i-th features of the p-th frame with the j-th neurons (see Figure 5.1), the SGD updates are computed with the following equation:

$$w_{p,i,j} = w_{p,i,j} - \eta \frac{\partial C}{\partial w_{p,i,j}} \tag{5.6}$$

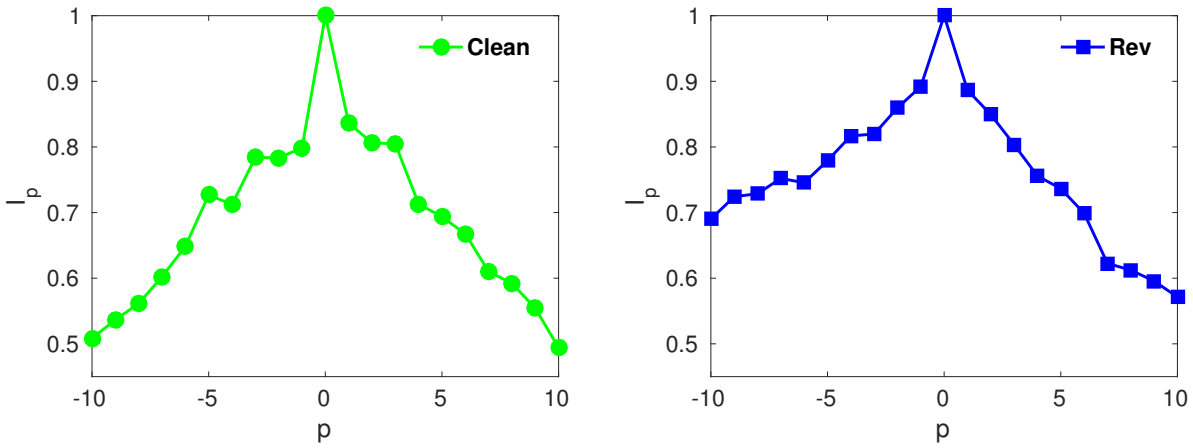
where η is the learning rate and C is the considered cost function (e.g., cross-entropy). Large gradients $\frac{\partial C}{\partial w_{p,i,j}}$ are typically attributed to impactful weights, whose little perturbation produces larger effects on the final cost function C. On the contrary, weights with little impact on the cost function will have, on average, smaller gradients and their values will not deviate significantly from their original close-to-zero initialization.

Relying on this characteristic, we can define the importance I_p for the p-th frame of the context window in the following way:

$$I_p = \sum_{i=1}^{N_{fea}} \sum_{j=1}^{N_{neu}} w_{p,i,j}^2 \tag{5.7}$$

where N_{fea} is the number of features for each frame, and N_{neu} is the number of neurons of the first hidden layer.

In this work we have first trained a DNN composed of six hidden layers with the WSJ dataset. Then, we computed the frame importance on the weights connecting the input features with the first hidden layer. This operation has been performed for both the standard close-talking version of the WSJ dataset, and for a corresponding reverberated version generated with impulse responses deriving from the DIRHA apartment (see App. C.4 for more details).



(a) Close-talking scenario (Clean) (b) Distant-talking scenario (Rev)

Figure 5.5: Frame Importance I_p obtained by analyzing the DNN weights connecting the various features frames of the context window.

The frame importance I_p is reported in Figure 5.5 for both the aforementioned close-talking (Figure 5.5a) and distant-talking conditions (Figure 5.5b). I_p is normalized by its maximum value for representational convenience. The index $p = 0$ represents the current frame, while negative and positive values of p refer to past and future frames, respectively.

This experiment highlights that the network, as expected, is able to automatically assign more importance to the current frame (p=0). In both close and distant-talking conditions the importance I_p clearly decreases when progressively moving far away from the current frame. Interestingly enough, in the reverberated case the network learns to place more importance to past frames ($p < 0$) rather than to future frames ($p > 0$). This can be readily appreciated from the asymmetric trend achieved in Figure 5.5b, which is a further indication of the possible benefits deriving from using asymmetric time contexts. In fact, with a proper asymmetric context, we can directly feed the network with the most important frames, avoiding to overload it with useless information. Note that a very different trend is obtained with close-talking data (see Figure 5.5a). In the latter case,

indeed, the network shows once again no clear preference for past or future information.

5.2.3 ASR experiments

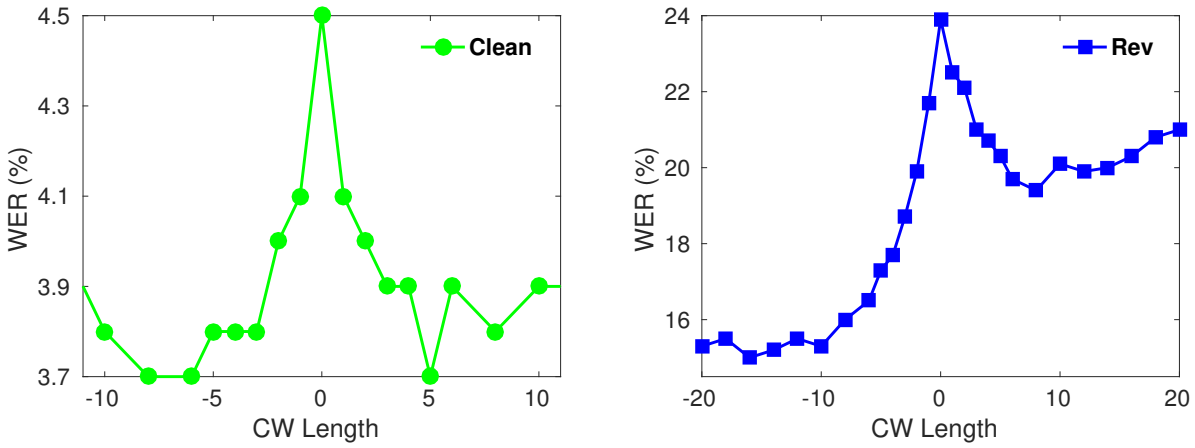
In the previous section, we found that the distant-talking DNN tends to naturally attribute more importance to past rather than future frames. In this section, we take a step forward by verifying whether this fact is observed even in terms of recognition performance.

The following part of this section reports the ASR experiments performed with different context window settings, features and architectures. We will also provide experimental evidence in different acoustic environments as well as under mismatching conditions.

The following experimental activity considered the WSJ dataset for training and the DIRHA-English-WSJ (set 1 portion) for test purposes. Close-talking, reverberated, and reverberated+noise versions of these corpora were considered. The experiments were based on a standard DNN composed of six hidden layers of 2048 neurons. Please, refer to the appendix for a more detailed description of the DIRHA-English-WSJ5k dataset and for the related baselines (App. B). For more information on the adopted experimental setup, please refer to App. C.4).

ASR performance with different context settings

As a first experiment, we examined the performance obtained in close-talking (Clean) and reverberated (Rev) conditions when fully asymmetric (i.e., single side) context windows are used. Figure 5.6 shows the outcome of these experiments, reporting results in terms of Word Error Rate (WER). Negative x-axis refers to the progressive integration of past frames only, while positive x-axis refers to future frames. In this set of experiments, fMLLR features are employed.



(a) Close-talking scenario (Clean) (b) Distant-talking scenario (Rev)

Figure 5.6: WER(%) obtained by DNN context windows that progressively integrate only past or future frames. Training is performed with a contaminated version of the WSJ dataset (WSJ-rev), while test is based on the DIRHA-English-WSJ5k (set1, simulated part).

Results highlight that in a close-talking framework a rather symmetric behaviour is attained (Figure 5.6a), reiterating that in such contexts past and future information provide a similar contribution to improve the system performance. Differently, the role of past information is significantly more important in a distant-talking case, since a faster decrease of the WER(%) is observed when past frames are progressively concatenated (Figure 5.6b). This result is in line with the findings emerged in the previous sections, and it confirms that an asymmetric context window is more suitable than a traditional symmetric one when reverberation arises.

In the latter experiment, we tested only fully asymmetric windows with ρ_{cw}=0% (future frames) or ρ_{cw} = 100% (past frames). However, it might be of interest to study more fuzzy configurations, where both past and future frames are considered. With this purpose, Figure 5.7 compares this kind of asymmetric windows with standard symmetric contexts of various lengths.

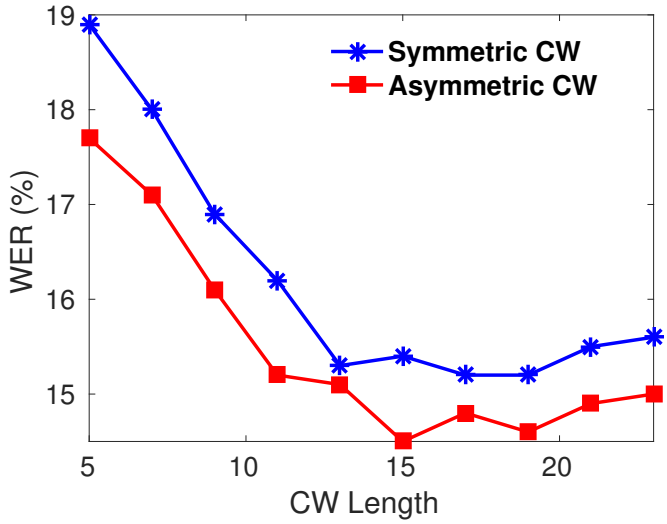


Figure 5.7: Comparison between symmetric and asymmetric windows of different durations in a reverberated distant-talking scenario (using the set1-rev of the DIRHA-English-WSJ5k).

From this experiment, it emerges that the asymmetric window consistently outperforms the standard symmetric one for all the considered context durations. On average, about 5% relative improvement of the WER is obtained with essentially no computational cost. This result refers to $\rho_{cw} = 60 \div 65\%$, which was optimal for all the context durations. Note that the optimal ρ_{cw} could also be predicted by simply analyzing the local minima emerged in Figure 5.6b, i.e., the points where no longer benefits are observed when adding new frames ($N_p = 11$, $N_f = 7$, $\rho_{cw} = 61\%$). The same trend is confirmed with other ρ_{cw} ranging from $55 \div 90\%$.

ASR performance with different features and architectures

To further validate the benefits of the asymmetric context window, we performed additional experiments with also FBANK and MFCC coefficients. Moreover we also studied the proposed context window on a standard CNN, that was composed of two convolutional layers followed by four

Architecture	Features	SCW (9-1-9)	ACW (11-1-7)
DNN	fMLLR	15.2	**14.8**
DNN	MFCC	21.8	**20.8**
DNN	FBANK	20.7	**20.2**
CNN	FBANK	18.5	**18.1**

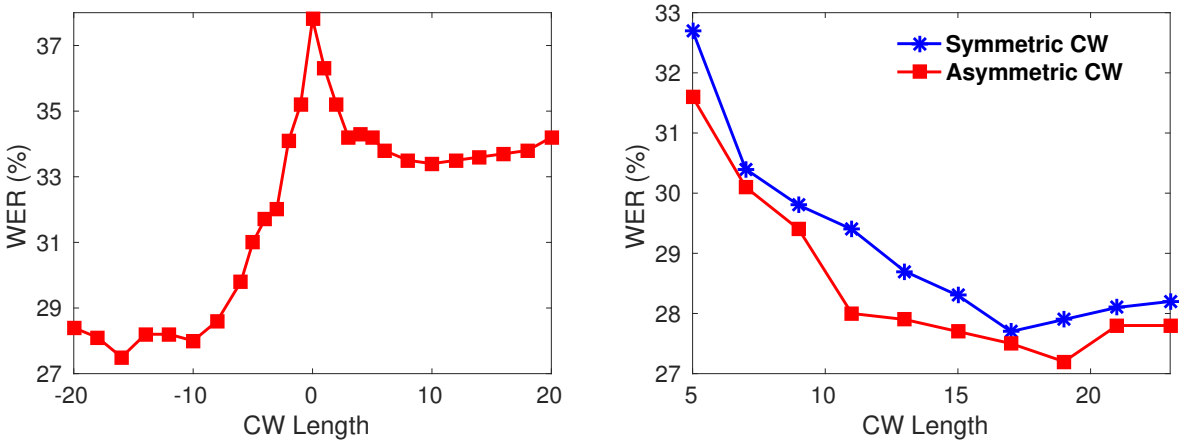
Table 5.1: Comparison between the WER(%) achieved with symmetric (SCW) and asymmetric context window (ACW) when different features and DNN architectures are used (using set1-rev of DIRHA-English-WSJ5k). The window configuration is reported in brackets with the following format: N_p-1-N_f.

fully-connected layers (see App. C.4 for more details).

Results of Table 5.1 confirm that the asymmetric context window outperforms the symmetric one in all the considered settings. Although the best performance is achieved with fMLLR features, the computation of such coefficients is not compliant with real-time constraints, since an additional decoding step performed with a HMM-GMM acoustic model is required. For real-time applications, standard MFCCs or FBANKs thus remain the most viable choice. The last row of Tab 5.1 highlights an interesting performance improvement achieved with CNNs. CNNs, which represent a valid alternative to fully-connected DNNs, are based on local connectivity, weight sharing, and pooling operations that allow them to exhibit some invariance to small feature shifts along the frequency axis, with well-known benefits against speaker and environment variations [1].

ASR performance under mismatching conditions

In the previous experiments, training and test have been performed on the same environment with similar acoustic conditions. In real applications, however, DSR systems often operate under mismatching conditions. As a first experiment to test the latter situation, we train the DNN with the reverberated data (Rev) so far considered, while test is performed with the



(a) Fully asymmetric context

(b) Symmetric vs asymmetric window

Figure 5.8: Comparison between symmetric and the proposed asymmetric context window under mismatching conditions. Training is performed with the WSJ-rev data, while test is performed on the set1-real (rev&noise) part of the DIRHA-English Dataset.

DIRHA-English data corrupted by both noise and reverberation (DIRHA-English-rev&noise). Figure 5.8a shows the results obtained when fully asymmetric context windows are adopted. Figure 5.8b, instead, compares symmetric and asymmetric time contexts of different duration with the optimal $\rho_{cw} = 60 \div 65$.

Due to the more challenging conditions, the WER(%) is significantly worse than that highlighted in Figure 5.6b and Figure 5.7. However, it is worth noting that the benefits deriving from using asymmetric contexts are maintained even under the addressed mismatching case.

Another mismatch typology might occur when training and test are performed in different acoustic environments. In the experiment reported in Table 5.2, training is carried out in the living-room of the DIRHA apartment so far considered, while test is performed in different environments, i.e., an office, a surgery room, as well as a room of another apartment. The test data were generated following our data simulation approach, but using impulse responses measured in the aforementioned environments. For

Environment	T_{60} (ms)	SCW (9-1-9)	ACW (11-1-7)
Office	650 ms	16.6	**16.2**
Home	700 ms	19.5	**19.1**
Surgery Room	850 ms	21.4	**20.3**

Table 5.2: WER(%) obtained with symmetric (SCW) and asymmetric (ACW) context windows in different acoustic environments and under mismatching conditions.

these experiments, we inherit a context length of 19 frames, that resulted optimal in previous studies.

Results reveal that all the various testing environments take benefit from the use of the asymmetric contexts, even when the training is performed in a different acoustic environment.

ASR performance with different reverberation times

As a last experiment, we further extend our validation by simulating acoustic environments with different reverberation times T_{60}. For this study, a set of impulse responses simulated with the directional image method described in Sec. 4.3 are used to contaminate both training (WSJ-clean) and testing corpora (DIRHA-WSJ-clean). Table 5.3 summarizes the optimal results obtained with T_{60} ranging from 250 ms to 1000 ms.

As expected, results show that the performance progressively degrades as T_{60} increases. The proposed solution, however, is able to overtake standard symmetric windows in all the different reverberation conditions considered here. It is also worth noting that larger context windows are needed when passing from simple acoustic conditions to more challenging environments. This derives from additional optimization experiments that showed, for instance, that when T_{60}=250 ms the optimal window integrates only 11 frames, while 25 frames are necessary when T_{60}=1000 ms. For the sake of completeness, the last column of Table 5.3 ($ACW*$) reports the results obtained when, contrary to what proposed in this work, a context window

T_{60}(ms)		SCW	ACW	ACW*
250 ms	CW Composition	*5-1-5*	*6-1-4*	*4-1-6*
	WER	5.5	**5.1**	5.7
500 ms	CW Composition	*6-1-6*	*7-1-5*	*5-1-7*
	WER	9.1	**8.5**	9.4
750 ms	CW Composition	*9-1-9*	*13-1-5*	*5-1-13*
	WER	15.2	**14.8**	16.8
1000 ms	CW Composition	*12-1-12*	*18-1-6*	*6-1-18*
	WER	20.5	**20.1**	23.2

Table 5.3: Comparison between WER(%) obtained with symmetric (SCW) and asymmetric (ACW) context window under different reverberation conditions. The ACW* column reports the results obtained with an asymmetric window that integrates more future than past frames.

embedding more future than past frames is considered. Results confirm once again that the latter choice is not optimal, since such a frame configuration integrates redundant information when the reverberation time increases. This degree of redundancy can also be related to the relative performance loss from ACW to $ACW*$, which is ranging from 10% to 15%.

In this section we studied asymmetric context windows that turned out to be helpful for improving DSR performance thanks to better management of the redundancy introduced by reverberation. A noteworthy aspect is that these benefits are obtained without introducing additional computational costs. Moreover, this approach minimizes the use of future frames, making it very suitable for practical small-footprint and on-line ASR applications. Although rather large contexts can be analyzed with this approach, one limitation lies in the relatively short-term analysis of speech modulations achievable with this methodology. The ASR performance, according to our experiments, starts to degrade when contexts larger than 150-200 ms are embedded.

In the following sections we focus on architectures able to learn longer-

term dependencies. We will initially recall the main state-of-the-art approaches and, finally, we propose a novel RNN architecture.

5.3 Managing Long-Term Dependencies

To embed longer-term information, some DNN architectures have been proposed in the past. Remaining in the domain of feed-forward DNNs, an example are Time Delay Neural Networks (TDNNs) [279, 200], that consider a wide fixed-size context window to manage long-range temporal dependencies with a modular and incremental architecture. In standard DNNs, all the neurons of the first hidden layer are fed with the entire temporal context. Differently, in TDNNs the initial transformations are learnt on narrow contexts and the deeper layers process the hidden activations from a wider temporal context.

Other popular approaches adopt a cascade of multiple DNNs for embedding larger contexts [157, 8, 269]. TempoRAl PatternS (TRAPS), for instance, capture long-term speech modulations by progressively analyzing long temporal vectors of critical band energies [115]. More precisely, the TRAPS architecture consists of two MLPs: the first one performs a non-linear mapping from log critical band energy time trajectories to phonetic probabilities, while the second MLP combines these predictions to obtain the overall phonetic probabilities. This method was able to successfully exploit contexts ranging from 500 ms to 1000 ms. A variant of TRAP is represented by the Hidden Activation TRAPS (HAT) [42, 304], where the first stage outputs hidden activations rather than posterior probabilities. A paradigm that recently gained popularity is Hierarchical DNNs (H-DNN) [102, 203, 217], H-DNNs are based on a first DNN focusing on short speech dependencies, while a second one works on a different time scale to capture long-term information. This approach was particularly

useful in the context of the BABEL project[1], where a bottleneck hidden layer in the first DNN was often used to derive language independent features [103, 102, 203].

In the first part of this PhD study, some efforts were devoted to study possible variations of this hierarchical paradigm. One contribution, described in [217], was the study of a novel H-DNN architecture, called multi-stream H-DNN. This variant, proposed as a joint work between FBK and ICSI-Berkeley, was tested and trained on spontaneous telephone speech conversations using the Cantonese IARPA Babel corpus and turned out to be effective for improving the ASR performance. A second contribution, reported in [218], was the extensions of this paradigm to an acoustic event classification task. To the best of our knowledge, our work, that was another result achieved in the context of the collaboration with ICSI, was the first exploring this paradigm in the latter domain. Please refer to the referenced paper for more details.

The feed-forward DNNs discussed so far are still very popular and represent the most preferable choice when the computational power is limited or when there are real-time constraints. A typical application is thus low-latency/real-time speech recognition or small-footprint systems, as witnessed by the numerous studies in the literature [161, 282, 43, 240, 119, 155]. Nevertheless, when enough computational capabilities are available, the most suitable architecture to process long-term information is represented by Recurrent Neural Networks (RNNs). Thanks to their recurrent nature, RNNs can memorize and process very long time contexts, and are therefore potentially able to outperform standard feed-forward DNNs. RNNs have recently gained a lot of popularity in speech recognition and have been recently used in the context of both hybrid RNN-HMM and end-to-end speech recognizers.

[1] `https://www.iarpa.gov/index.php/research-programs/babel`

At the time of writing, the most popular RNN architecture able to learn long-term dependencies is LSTM, that exploits multiplicative gates to create gradient shortcuts. General-purpose RNNs such as Long Short Term Memories (LSTMs) [118] have been the subject of several studies and modifications over the past years [85, 100, 302], leading to some novel architectures, such as the recently-proposed Gated Recurrent Units (GRUs) [46]. Note that the asymmetric context window studied in the previous section, can also be used for modern GRU or LSTM architectures. However, some additional experiments not reported here, suggest that no benefits can be obtained with this approach. On the other hand, RNNs automatically learn how to exploit the time information through the recurrent connections, and the adoption of a context window is rather redundant and useless.

Instead, our work, that will be summarized in the next sections, attempts to improve the context management by further revising GRUs. Differently from previous efforts, our primary goal was not to derive a general-purpose RNN, but to modify the standard GRU design to better process time information for speech recognition, in particular for noisy and reverberant speech inputs.

5.4 Revising Gated Recurrent Units

LSTMs rely on memory cells that are controlled by forget, input, and output gates. Despite their effectiveness, such a sophisticated gating mechanism might result in an overly complex model.

A noteworthy attempt to simplify LSTMs has recently led to a novel model called Gated Recurrent Unit (GRU) [46, 47], that is based on just two multiplicative gates. In particular, the standard GRU architecture is

defined by the following equations:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z), \tag{5.8a}$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r), \tag{5.8b}$$

$$\widetilde{h_t} = \tanh(W_h x_t + U_h(h_{t-1} \odot r_t) + b_h), \tag{5.8c}$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \widetilde{h_t}. \tag{5.8d}$$

where z_t and r_t are vectors corresponding to the update and reset gates, respectively, while h_t represents the state vector for the current time frame t. Computations denoted as $\odot$ represent element-wise multiplications. The activations of both gates are element-wise logistic sigmoid functions $\sigma(\cdot)$, that constrain z_t and r_t to take values ranging from 0 and 1. The candidate state $\widetilde{h_t}$ is processed with a hyperbolic tangent. The network is fed by the current input vector x_t (e.g., a vector of speech features), while the parameters of the model are the matrices W_z, W_r, W_h (the feed-forward connections) and U_z, U_r, U_h (the recurrent weights). The architecture finally includes trainable bias vectors b_z, b_r and b_h, that are added before the non-linearities are applied.

As shown in Eq. 5.8d, the current state vector h_t is a linear interpolation between the previous activation h_{t-1} and the current candidate state $\widetilde{h_t}$. The weighting factors are set by the update gate z_t, that decides how much the units will update their activations. Note that this linear interpolation, which is similar to the gating in LSTMs [118], is the key component for learning long-term dependencies. For instance, if z_t is close to one, the previous state is kept unaltered and can remain unchanged for an arbitrary number of time steps. On the other hand, if z_t is close to zero, the network tends to favor the candidate state $\widetilde{h_t}$, that depends more heavily on the current input and on the closer hidden states. The candidate state $\widetilde{h_t}$ also depends on the reset gate r_t, that allows the model to possibly delete the

past memory by forgetting the previously computed states.

Despite the interesting performance achieved by GRUs that, according to several works in the literature [47, 131, 124], is comparable to LSTMs in different machine learning tasks, in this thesis we tried to further simplify this standard model, deriving a GRU architecture able to better process long speech contexts. In particular, the main changes to the standard GRU model concern the reset gate, ReLU activations, and batch normalization, as outlined in the next sub-sections.

5.4.1 Removing the reset gate

From the previous introduction to GRUs, it follows that the reset gate can be useful when significant discontinuities occur in the sequence. For language modeling, this may happen when moving from one text to another that is not semantically related. In such situation, it is convenient to reset the stored memory to avoid taking a decision biased by an unrelated history.

Nevertheless, we believe that for some specific tasks like speech recognition this functionality might not be useful. In fact, a speech signal is a sequence that evolves rather slowly (the features are typically computed every 10 ms), in which the past history can virtually always be helpful. Even in the presence of strong discontinuities, for instance observable at the boundary between a vowel and a fricative, completely resetting the past memory can be harmful. On the other hand, it is helpful to memorize phonotactic features, since some phone transitions are more likely than others.

We also argue that a certain redundancy in the activations of reset and update gates might occur when processing speech sequences. For instance, when it is necessary to give more importance to the current information, the GRU model can set small values of r_t. A similar effect can be achieved

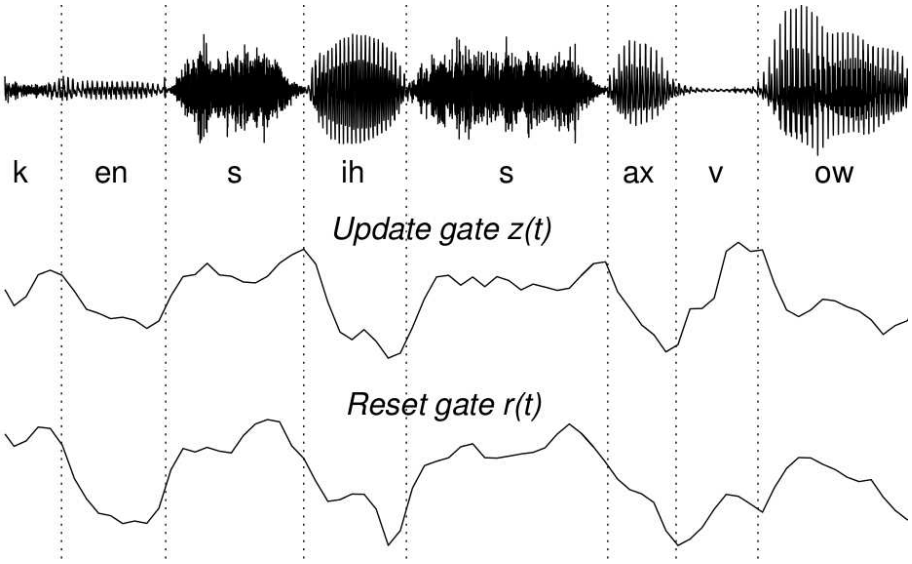


Figure 5.9: Average activations of the update and reset gates for a GRU trained on TIMIT in a chunk of the utterance "*sx403*" of speaker "*faks0*". The activations are within the rage 0.7÷0.3 for both gates.

with the update gate only, if small values are assigned to z_t. The latter solution tends to weight more the candidate state $\widetilde{h}_t$, that depends heavily on the current input. Similarly, a high value can be assigned either to r_t or to z_t, in order to place more importance on past states. This redundancy is also highlighted in Figure 5.9, where a temporal correlation in the average activations of update and reset gates can be readily appreciated for a GRU trained on TIMIT.

Based on these reasons, the first variation to standard GRUs thus concerns the removal of the reset gate r_t. This change leads to the following modification of Eq. 5.8c:

$$\widetilde{h}_t = \tanh(W_h x_t + U_h h_{t-1} + b_h) \tag{5.9}$$

The main benefits of this intervention are related to the improved computational efficiency, that is achieved thanks to a more compact single-gate model. Note that the removal of the reset gate does not prevent the

architecture to learn long-term dependencies. The weights of the linear interpolation, in fact, are determined by the update gate only, and even avoiding the reset gate, the aforementioned shortcut used to mitigate gradient vanishing problems is still preserved.

5.4.2 ReLU activations

As outlined in Chapter 2, ReLU activations are very popular for feed-forward DNNs. We believe that extending the well-known benefits of ReLU to RNNs is of great importance, since it might help mitigating vanishing gradient issues and fostering a better learning of long-term dependencies.

In this work, our second modification thus consists in replacing the standard hyperbolic tangent of standard GRU with ReLU activation. In particular, we modify the computation of candidate state $\widetilde{h}_t$ (Eq. 5.8c), as follows:

$$\widetilde{h}_t = \mathrm{ReLU}(W_h x_t + U_h h_{t-1} + b_h) \tag{5.10}$$

The adoption of ReLU-based neurons was not so common in the past for the RNN architectures. This was due to numerical instabilities originating from the unbounded ReLU functions applied over long time series. In the proposed architecture we were able to circumvent these numerical issues coupling it with batch normalization, as will be discussed in the next subsection.

5.4.3 Batch Normalization

Batch normalization [123] has been recently proposed in the machine learning community and addresses the so-called *internal covariate shift* problem by normalizing the mean and the variance of each layer's pre-activations for each training mini-batch. Several works have already shown that this

technique is effective both to improve the system performance and to speed-up the training procedure [148, 4, 212, 215]. Batch normalization can be applied to RNNs in different ways. In [148], the authors suggest to apply it to feed-forward connections only, while in [49] the normalization step is extended to recurrent connections, using separate statistics for each time-step. In our work, we tried both approaches, but we did not observe substantial benefits when extending batch normalization to recurrent parameters (i.e., U_h and U_z). For this reason, we applied this technique to feed-forward connections only (i.e., W_h and W_z), obtaining a more compact model that is almost equally performing but significantly less computationally expensive. When batch normalization is limited to feed-forward connections, indeed, all the related computations become independent at each time step and they can be performed in parallel. This offers the possibility to apply it with reduced computational efforts. As outlined in the previous sub-section, coupling the proposed model with batch-normalization [123] could also help in limiting the numerical issues of ReLU RNNs. Batch normalization, in fact, rescales the neuron pre-activations, inherently bounding the values of the ReLU neurons. In this way, our model concurrently takes advantage of the well-known benefits of both ReLU activation and batch normalization. In our experiments, we found that the latter technique helps against numerical issues also when it is limited to feed-forward connections only.

Formally, removing the reset gate, replacing the hyperbolic tangent function with the ReLU activation, and applying batch normalization, now

leads to the following model:

$$z_t = \sigma(BN(W_z x_t) + U_z h_{t-1}), \tag{5.11a}$$

$$\widetilde{h_t} = \text{ReLU}(BN(W_h x_t) + U_h h_{t-1}), \tag{5.11b}$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \widetilde{h_t}. \tag{5.11c}$$

The batch normalization $BN(\cdot)$ works as described in [123], and is defined as follows:

$$BN(a) = \gamma \odot \frac{a - \mu_b}{\sqrt{\sigma_b^2 + \epsilon}} + \beta \tag{5.12}$$

where μ_b and σ_b are the minibatch mean and variance, respectively. A small constant ϵ is added for numerical stability. The variables γ and β are trainable scaling and shifting parameters, introduced to restore the network capacity. Note that the presence of β makes the biases b_h and b_z redundant. Therefore, they are omitted in Eq. 5.11a and 5.11b.

We called this architecture Light GRU (Li-GRU), to emphasize the simplification process conducted on a standard GRU.

5.4.4 Related work

A first attempt to remove r_t from GRUs has recently led to a single-gate architecture called Minimal Gated Recurrent Unit (M-GRU) [303], that achieves a performance comparable to that obtained by standard GRUs in handwritten digit recognition as well as in a sentiment classification task. To the best of our knowledge, our contribution is the first attempt that explores this architectural variation in speech recognition. Recently, some attempts have also been done for embedding ReLU units in the RNN framework. For instance, in [149] authors replaced tanh activations with ReLU neurons in a vanilla RNN, showing the capability of this model to

learn long-term dependencies when a proper orthogonal initialization is adopted. In this work, we extend the use of ReLU to a GRU architecture.

In summary, the novelty of our approach consists in the integration of three key design aspects (i.e, the removal of the reset gate, ReLU activations and batch normalization) in a single model, that resulted particularly suitable for speech recognition.

In the following sub-sections, we describe the experimental activity conducted to assess the proposed model. Most of the experiments reported in the following are based on hybrid DNN-HMM speech recognizers, since the latter ASR paradigm typically reaches state-of-the-art performance. However, for the sake of comparison, we also extended the experimental validation to an end-to-end CTC model. More precisely, in sub-section 5.4.5, we first quantitatively analyze the correlations between the update and reset gates in a standard GRU. In sub-section 5.4.6, we extend our study with some analysis of gradient statistics. The role of batch normalization and the CTC experiments are described in sub-sections 6.4.2 and 5.4.8, respectively. The speech recognition performance will then be reported for TIMIT, DIRHA-English-WSJ, CHiME as well as for the TED-talk corpus. The complete experimental setup is described in App. C.5, where more details about DNN and ASR setups are reported.

5.4.5 Correlation analysis

The degree of redundancy between reset and update gates can be analyzed in a quantitative way using the cross-correlation metric $C(z, r)$:

$$C(z, r) = \overline{z}_t \star \overline{r}_t \tag{5.13}$$

where $\overline{z}_t$ and $\overline{r}_t$ are the average activations (over the neurons) of update and reset gates, respectively, and $\star$ is the cross-correlation operator.

The cross-correlation $C(z, r)$ between the average activations of update

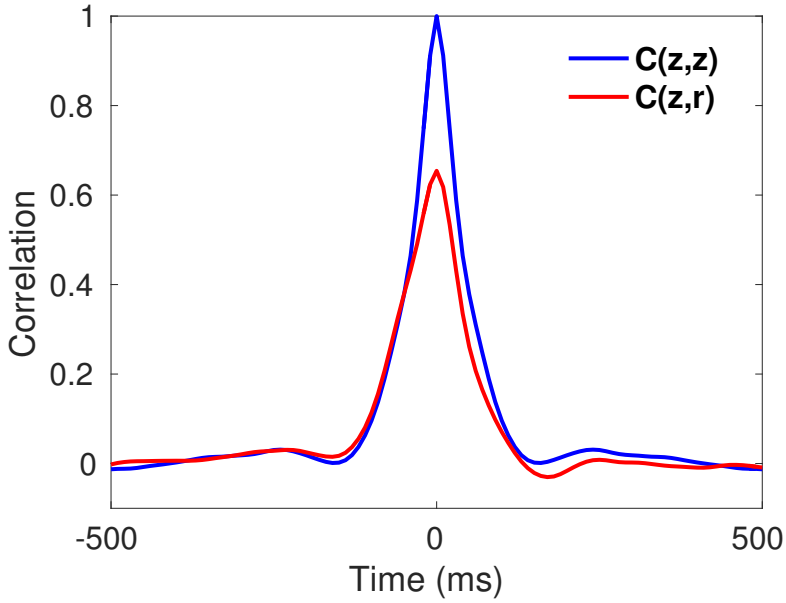


Figure 5.10: Auto-correlation $C(z, z)$ and cross-correlation $C(z, r)$ between the average activations of the update (z) and reset (r) gates. Correlations are normalized by the maximum of $C(z, z)$ for graphical convenience.

z and reset r gates is shown in Figure 5.10 for a standard GRU trained on the TIMIT dataset (see App. B). The gate activations are computed for all the input frames and, at each time step, an average over the hidden neurons is considered. The cross-correlation $C(z, r)$ is displayed along with the auto-correlation $C(z, z)$, that represents the upper-bound limit of the former function.

Figure 5.10 clearly shows a high peak of $C(z, r)$, revealing that update and reset gates end up being redundant. This peak is about 66% of the maximum of $C(z, z)$ and it is centered at $t = 0$, indicating that almost no-delay occurs between gate activations. This result is obtained with a single-layer GRU of 200 bidirectional neurons fed with MFCC features and trained with TIMIT. After the training-step, the cross-correlation is averaged over all the development sentences.

It would be of interest to examine the evolution of this correlation over

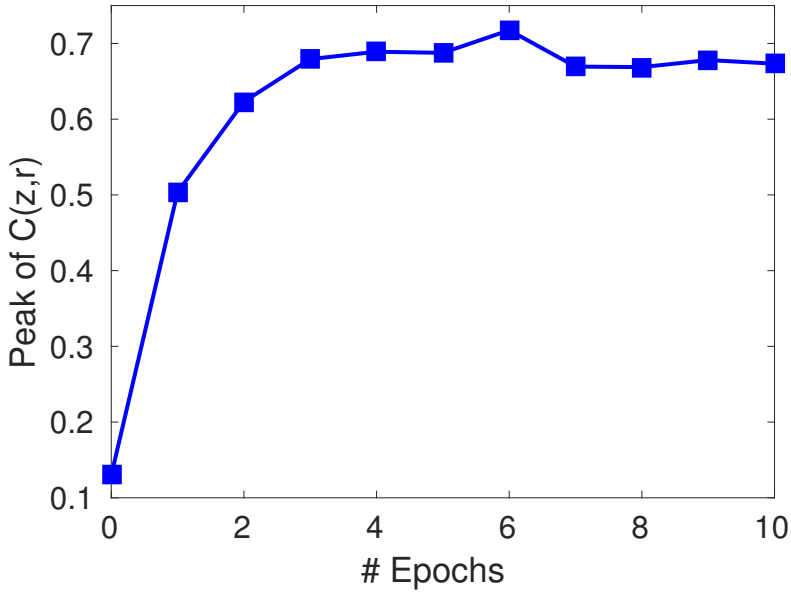


Figure 5.11: Evolution of the peak of the cross-correlation $C(z, r)$ over various training epochs.

the epochs. With this regard, Figure 5.11 reports the peak of $C(z, r)$ for some training epochs, showing that the GRU attributes rather quickly a similar role to update and reset gates. In fact, after 3-4 epochs, the correlation peak reaches its maximum value, that is almost maintained for all the subsequent training iterations.

5.4.6 Gradient analysis

The analysis of the main gradient statistics can give some preliminary indications on the role played by the various parameters.

With this goal, Table 5.4 reports the L2 norm of the gradient for the main parameters of the GRU model considered in the previous section. Results reveal that the reset gate weight matrices (i.e., W_r and U_r) have a smaller gradient norm when compared to the other parameters. This result is somewhat expected, since the reset gate parameters are processed by two different non-linearities (i.e., the sigmoid of Eq. 5.8b and the tanh

Param. \ *Arch.*	GRU	M-GRU	Li-GRU
$\|W_h\|$	4.80	4.84	4.86
$\|W_z\|$	0.85	0.89	0.99
$\|W_r\|$	0.22	-	-
$\|U_h\|$	0.15	0.35	0.71
$\|U_z\|$	0.05	0.11	0.13
$\|U_r\|$	0.02	-	-

Table 5.4: L_2 norm of the gradient for the main parameters of the GRU models. The norm is averaged over all the training sentences and epochs.

of 5.8c), that can attenuate their gradients. This would anyway indicate that, on average, the reset gate has less impact on the final cost function, further supporting its removal from the GRU design. When avoiding it, the norm of the gradient tends to increase (see for instance the recurrent weights U_h of M-GRU model). This suggests that the functionalities of the reset gate, can be performed by other model parameters. The norm further increases in the case of Li-GRUs, due to the adoption of ReLu units. This non-linearity, indeed, improves the back-propagation of the gradient over both time-steps and hidden layers, making long-term dependencies easier to learn. Results are obtained with the same GRU used in subsection 5.4.5, considering M-GRU and Li-GRU models with the same number of hidden units.

5.4.7 Impact of batch normalization

After the preliminary analysis on correlation and gradients done in previous sub-sections, we now compare RNN models in terms of their final speech recognition performance on the TIMIT dataset. In particular, this section studies the effects of batch normalization on the proposed model. With this regard, Table 5.5 compares the Phone Error Rate (PER%) achieved

Param. \ *Arch.*	GRU	M-GRU	Li-GRU
without batch norm	18.4	18.6	20.4
with batch norm	17.1	17.2	**16.7**

Table 5.5: PER(%) of the GRU models with and without batch normalization (TIMIT dataset, MFCC features).

with and without this technique.

Results show that batch normalization is helpful to improve the ASR performance, leading to a relative improvement of about 7% for GRU and M-GRU and 18% for the proposed Li-GRU. The latter improvement confirms that our model couples particularly well with this technique, due to the adopted ReLU activations. Without batch normalization, the ReLU activations of the Li-GRU are unbounded and tend to cause numerical instabilities. According to our experience, the convergence of Li-GRU without batch normalization, can be achieved only by setting rather small learning rate values. The latter setting, however, can lead to a poor performance and, as clearly emerged from this experiment, coupling Li-GRU with this technique is strongly recommended.

5.4.8 CTC experiments

The proposed Li-GRU model has also been evaluated in the context of end-to-end speech recognition, considering the CTC technique introduced in Chapter 3. Table 5.6 summarizes the results of CTC on the TIMIT data set. In these experiments, the Li-GRU clearly outperforms the standard GRU, showing the effectiveness of the proposed model even in a end-to-end ASR framework. The improvement is obtained both with and without batch normalization and, similarly to what observed for hybrid systems, the latter technique leads to better performance when coupled with Li-GRU.

Arch. \ *Batch-norm.*	False	True
GRU	22.1	22.0
Li-GRU	21.1	**20.9**

Table 5.6: PER(%) obtained for the test set of TIMIT with various CTC RNN architectures.

However, a smaller performance gain is observed when batch normalization is applied to the CTC. This result could also be related to the different choice of the regularizer, as weight noise was used instead of recurrent dropout.

In general, PERs are higher than those of the hybrid systems. End-to-end methods, in fact, are relatively young models, that are still less competitive than more complex (and mature) DNN-HMM approaches. We believe that the gap between CTC and hybrid speech recognizers could be partially reduced in our experiments with a more careful setting of the hyperparameters and with the introduction of an external phone-based language model. The main focus of this work, however, is to show the effectiveness of the proposed Li-GRU model, and a fair comparison between CTC and hybrid systems is out of the scope of this work.

5.4.9 DNN-HMM Experiments

The results of Table 5.5 and 5.6 highlighted that the proposed Li-GRU model outperforms other GRU architectures. In this sub-section, we extend this study by performing a more detailed comparison with the most popular RNN architectures. To provide a fair comparison, batch normalization is hereinafter applied to all the considered RNN models. Moreover, at least five experiments varying the initialization seeds were conducted for each RNN architecture. The results are thus reported as the average error rates

Arch. \ *Feat.*	MFCC	FBANK	fMLLR
relu-RNN	18.7 ± 0.18	18.3 ± 0.23	16.3 ± 0.11
LSTM	18.1 ± 0.33	17.1 ± 0.36	15.7 ± 0.32
GRU	17.1 ± 0.20	16.7 ± 0.36	15.3 ± 0.28
M-GRU	17.2 ± 0.11	16.7 ± 0.19	15.2 ± 0.10
Li-GRU	**16.7** ± 0.26	**15.8** ± 0.10	**14.9** ± 0.27

Table 5.7: PER(%) obtained for the test set of TIMIT with various RNN architectures.

with their corresponding standard deviation. In the following sub-sections, the results obtained for TIMIT, DIRHA-English-WSJ, CHIME and TED-Talk datasets are reported.

Recognition performance on TIMIT

Table 5.7 presents a comparison of the performance achieved with the most popular RNN models on the standard TIMIT dataset. The first row reports the results achieved with a simple RNN with ReLU activations (no gating mechanisms are used here). Although this architecture has recently shown promising results in some machine learning tasks [149], our results confirm that gated recurrent networks (rows 2-5) outperform traditional RNNs. We also observe that GRUs tend to slightly outperform the LSTM model. As expected, M-GRU (i.e., the architecture without reset gate) achieves a performance very similar to that obtained with standard GRUs, further supporting our speculation on the redundant role played by the reset gate in a speech recognition application. The last row reports the performance achieved with the proposed model, in which, besides removing the reset gate, ReLU activations are used. The Li-GRU performance indicates that our architecture consistently outperforms the other RNNs over all the considered input features. A remarkable achievement is the average PER(%) of 14.9% obtained with fMLLR features. To the best of

Phonetic Cat.	Phone Lists	GRU	Li-GRU
Vowels	{*iy,ih,eh,ae,...,oy,aw,ow,er*}	23.2	**23.0**
Liquids	{*l,r,y,w,el*}	20.1	**19.0**
Nasals	{*en,m,n,ng*}	16.8	**15.9**
Fricatives	{*ch,jh,dh,z,v,f,th,s,sh,hh,zh*}	17.6	**17.0**
Stops	{*b,d,g,p,t,k,dx,cl,vcl,epi*}	18.5	**17.9**

Table 5.8: PER(%) of the TIMIT dataset (MFCC features) split into five different phonetic categories. Silences (*sil*) are not considered here.

Architecture	Layers	Neurons	# Params
relu-RNN	4	607	6.1 M
LSTM	5	375	8.8 M
GRU	5	465	10.3 M
M-GRU	5	465	7.4 M
Li-GRU	5	465	7.4 M

Table 5.9: Optimal number of layers and neurons for each TIMIT RNN model. The outcome of the optimization process is similar for all considered features.

our knowledge, this result yields the best published performance on the TIMIT test-set.

In Table 5.8 the PER(%) performance is split into five different phonetic categories (vowels, liquids, nasals, fricatives and stops), showing that Li-GRU exhibits the best results for all the considered classes.

Previous results are obtained after optimizing the main hyperparameters of the model on the development set. Table 5.9 reports the outcome of this optimization process, with the corresponding best architectures obtained for each RNN architecture. For GRU models, the best performance is achieved with 5 hidden layers of 465 neurons. It is also worth noting that M-GRU and Li-GRU have about 30% fewer parameters compared to the standard GRU.

Arch. \ *Feat.*	MFCC	FBANK	fMLLR
relu-RNN	29.7 ± 0.31	30.0 ± 0.38	24.7 ± 0.28
LSTM	29.5 ± 0.41	29.1 ± 0.42	24.6 ± 0.35
GRU	28.5 ± 0.37	28.4 ± 0.21	24.0 ± 0.27
M-GRU	28.4 ± 0.34	28.1 ± 0.30	23.6 ± 0.21
Li-GRU	**27.8** ± 0.38	**27.6** ± 0.36	**22.8** ± 0.26

Table 5.10: Word Error Rate (%) obtained with the DIRHA English WSJ dataset (simulated part of the set2 rev&noise portion) for various RNN architectures.

Arch. \ *Feat.*	MFCC	FBANK	fMLLR
relu-RNN	23.7 ± 0.21	23.5 ± 0.30	18.9 ± 0.26
LSTM	23.2 ± 0.46	23.2 ± 0.42	18.9 ± 0.24
GRU	22.3 ± 0.39	22.5 ± 0.38	18.6 ± 0.23
M-GRU	21.5 ± 0.43	22.0 ± 0.37	18.0 ± 0.21
Li-GRU	**21.3** ± 0.38	**21.4** ± 0.32	**17.6** ± 0.20

Table 5.11: Word Error Rate (%) obtained with the DIRHA-English-WSJ dataset (real part of the set2 rev&noise portion) for various RNN architectures.

Recognition performance on DIRHA-English-WSJ

After a first set of experiments on TIMIT, in this sub-section we assess our model on a more challenging and realistic distant-talking task, using the DIRHA-English-WSJ corpus. A challenging aspect of this dataset is the acoustic mismatch between training and testing conditions. Training, in fact, is performed with a reverberated version of WSJ (WSJ-rev), while test is characterized by both non-stationary noises and reverberation.

Tables 5.10 and 5.11 summarize the results obtained with the simulated and real parts of this dataset.

These results exhibit a trend comparable to that observed for TIMIT, confirming that Li-GRU still outperforms GRU even in a more challenging

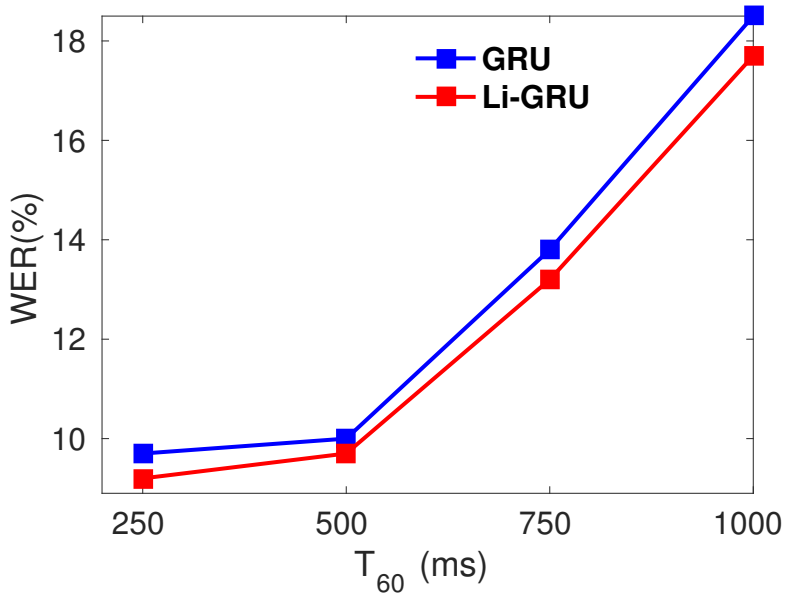


Figure 5.12: Evolution of the WER(%) for the DIRHA WSJ simulated data over different reverberation times T_{60}.

scenario. The results are consistent over both real and simulated data as well as across the different features considered in this study.

The reset gate removal seems to play a more crucial role in the addressed distant-talking scenario. If the close-talking performance reported in Table 5.7 highlights comparable error rates between standard GRU and M-GRU, in the distant-talking case we even observe a small performance gain when removing the reset gate. We suppose that this behaviour is due to reverberation, that implicitly introduces redundancy in the signal, due to the multiple delayed replicas of each sample. This results in a forward memory effect, that can make reset gate ineffective.

In Figure 5.12, we extend our previous experiments by generating simulated data with different reverberation times T_{60} ranging from 250 to 1000 ms, as outlined in Sec. 4.3. In order to simulate a more realistic situation, different impulse responses have been used for training and testing purposes. No additive noise is considered for these experiments.

Arch. \ *Dataset*	DT-sim	DT-real	ET-sim	ET-real
DNN	17.8 ± 0.38	16.1 ± 0.31	26.1 ± 0.45	30.0 ± 0.48
DNN+sMBR	14.7 ± 0.25	15.7 ± 0.23	24.0 ± 0.31	27.0 ± 0.35
GRU	15.8 ± 0.30	14.8 ± 0.25	23.0 ± 0.38	23.3 ± 0.35
M-GRU	15.9 ± 0.32	14.1 ± 0.31	22.8 ± 0.39	23.0 ± 0.41
Li-GRU	**13.5** ± 0.25	**12.5** ± 0.22	**20.3** ± 0.31	**20.0** ± 0.33

Table 5.12: Speech recognition performance on the CHiME dataset (single channel, fMLLR features).

As expected, the performance degrades as the reverberation time increases. Similarly to previous achievements, we still observe that Li-GRU outperforms GRU under all the considered reverberation conditions.

Recognition performance on CHiME

In this sub-section we extend the results to the CHiME corpus, that is an important benchmark in the ASR field, thanks to the success of CHiME challenges [11, 10]. In Table 5.12 a comparison across the various GRU architectures is presented. For the sake of comparison, the results obtained with the official CHiME 4 are also reported in the first two rows[2].

Results confirm the trend previously observed, highlighting a significant relative improvement of about 14% achieved when passing from GRU to the proposed Li-GRU. Similarly to our findings of the previous section, some small benefits can be observed when removing the reset gate. The largest performance gap, however, is reached when adopting ReLU units (see M-GRU and Li-GRU columns), confirming the effectiveness of this architectural variation. Note also that the GRU systems significantly outperform the DNN baseline, even when the latter is based on sequence

[2]The results obtained in this section are not directly comparable with the best systems of the CHiME 4 competition. Due to the purpose of this work, indeed, techniques such as multi-microphone processing, data-augmentation, system combination as well as lattice rescoring are not used here.

Arch. \ *Env.*	BUS	CAF	PED	STR
DNN	44.1	32.0	26.2	17.7
DNN+sMBR	40.5	28.3	22.9	16.3
GRU	33.5	25.6	19.5	14.6
M-GRU	33.1	24.9	19.2	14.9
Li-GRU	**28.0**	**22.1**	**16.9**	**13.2**

Table 5.13: Comparison between GRU and Li-GRU for the four different noisy conditions considered in CHiME on the real evaluation set (ET-real).

discriminative training (DNN+sMBR)[273].

Table 5.13 splits the ASR performance of the real test set into the four noisy categories. Li-GRU outperforms GRU in all the considered environments, with a performance gain that is higher when more challenging acoustic conditions are met. For instance, we obtain a relative improvement of 16% in the bus (BUS) environment (the noisiest), against the relative improvement of 9.5% observed in the street (STR) recordings.

Recognition performance on TED-talks

Table 5.14 reports a comparison between GRU and Li-GRU on the TED-talks corpus. The experiments are performed with standard MFCC features, and a four-gram language model is considered in the decoding step (see [236] for more details).

Results on both test sets consistently shows the performance gain achieved with the proposed architecture. This further confirms the effectiveness of Li-GRU, even for a larger scale ASR task. In particular, a relative improvement of about 14-17% is achieved. This improvement resulted statistically significant according to the matched-pair test [88] (conducted with NIST sclite *sc_stat* tool).

Arch. \ *Dataset.*	TST-2011	TST-2012
GRU	16.3	17.0
Li-GRU	**14.0**	**14.6**

Table 5.14: Comparison between GRU and Li-GRU with the TED-talks corpus.

Arch. \ *Dataset.*	TIMIT	DIRHA	CHiME	TED
GRU	9.6 min	40 min	312 min	590 min
Li-GRU	**6.5 min**	**25 min**	**205 min**	**447 min**

Table 5.15: Per-epoch training time (in minutes) of GRU and Li-GRU models for the various datasets on an NVIDIA K40 GPU.

Training time comparison

In the previous subsections, we reported several speech recognition results, showing that Li-GRU outperforms other RNNs. In this sub-section, we finally focus on another key aspect of the proposed architecture, namely its improved computational efficiency. In Table 5.15, we compare the per-epoch wall-clock training time of GRU and Li-GRU models.

The training time reduction achieved with the proposed architecture is about 30% for all the datasets. This reduction reflects the amount of parameters saved by Li-GRU, that is also around 30%. The reduction of the computational complexity, originated by a more compact model, also arises for testing purposes, making our model potentially suitable for small-footprint ASR, [161, 282, 43, 240, 119, 155], which studies DNNs designed for portable devices with small computational capabilities.

5.5 Summary and Future Challenges

This chapter has confirmed the considerable importance of time contexts for improving distant speech recognition. In our studies we considered both feed-forward and recurrent recurrent neural networks. For the former architecture we proposed asymmetric context windows, that turned out to be very precious to mitigate the effects of acoustic reverberation. Our efforts have then be devoted to revise standard GRUs, proposing a novel architecture, called Light GRU, that improves the ASR performance while reducing the training time of more that 30%.

Despite our contribution, we believe that there is still room to improve current techniques for managing time contexts. As discussed in this Chapter, the analysis of long-term dependencies is tightly connected with the problem of vanishing gradients over long computational chains, and the dominant approach consists in devising architectures with proper gradient shortcuts. Although this solution currently leads to interesting performance levels, it does not necessarily mean that this is the only possible methodology to embed long-term information. We indeed believe that, in the future, alternative architectures or novel training algorithms could play an important role for improving current techniques.

Moreover, there are other open challenges that state-of-the-art architectures are not able to address properly. For instance, a feature that is currently missing even in modern RNN implementations is the ability to recover the deleted memory. When processing long time sequences, the network can, at a certain point, erroneously erase the past memory, influencing the processing of the following elements of the sequence, that cannot benefit anymore from longer context. Solutions able to mitigate this issue could have a remarkable impact on distant speech recognition.

Chapter 6

Cooperative Networks of Deep Neural Networks

The current development of deep learning draws inspiration from multiple sources [92]. Among the others, the most important ones are the biological brain, game theory and natural evolution. According to Darwin's theories, a key role for the evolution of living forms is played by competition. Actually, competition across deep neural networks has been recently explored in the context of Generative Adversarial Networks (GAN) [93], where a generator and a discriminator continuously evolve themselves to improve their performance. Competition was also explored in deep reinforcement learning. Super-human performance, for instance, has been achieved in automated game playing, by forcing a competition between agents [67]. With this approach, DeepMind stunned the world when the AlphaGo system was able to defeat the world champion of Go, an ancient Chinese game with a huge number of possible moves (enormously larger than the combinations of chess).

Beyond competition, we believe that another key aspect to consider is cooperation [171]. Building neural networks able to automatically learn how to cooperate and communicate will represent a fundamental step towards artificial intelligence, that promises to bridge the gap between cur-

rent neural networks and human brain.

Inspired by this vision we developed a novel deep learning paradigm, called network of deep neural network. This paradigm can be exploited to solve challenging problems, where the cooperation across different DNNs can be helpful to counteract uncertainty. Distant speech recognition represents the natural application field for this approach: DSR, in fact, is not only a challenging multi-disciplinary problem, but it also requires a proper matching, communication and cooperation across the various modules involved in the speech recognition process. In the previous chapters of this thesis we tried to counteract the uncertainty originated by noise and reverberation using contaminated data or exploiting time contexts. Conversely, in the following we summarize our efforts to mitigate the harmful effects of these disturbances with another strategy: cooperation across DNNs.

The remaining part of this Chapter is organized as follows: the general paradigm will be described in Sec. 6.1, while its application to DSR is discussed in Sec. 6.2. The first experiments on joint training will be presented in Sec. 6.3, while the experimental evidence emerged with the final network of DNNs is reported in Sec. 6.4.

6.1 General Paradigm

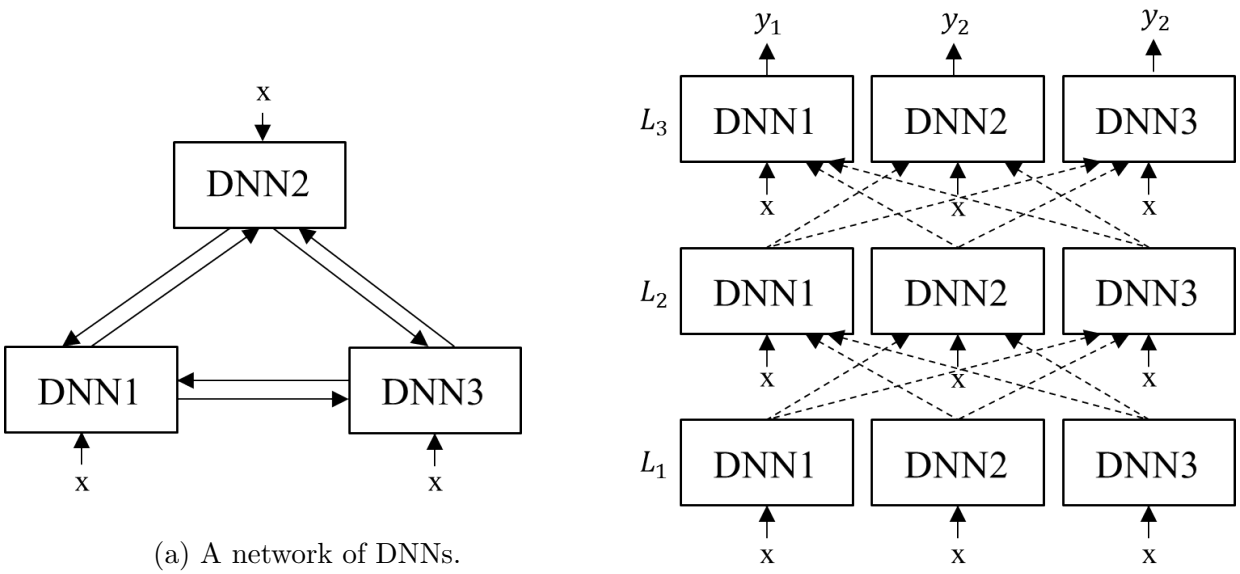
The human ability to solve complex problems is a distinctive trait of our species that contributes more than other abilities to the development of human life on earth. The typical approach to solve complex problems is to break them down into a series of simpler sub-problems. From an engineering point of view, this means that systems able to solve challenging tasks are often based on complex architectures. These architectures are composed of an ecosystem of smaller sub-modules, each one with its own functionality. A noteworthy example is our brain, in which different areas

specialized on different tasks communicate together to achieve complex goals [25].

The ability to solve multiple complex tasks through cooperation represents one of the most significant gaps between biological and artificial neural networks. Current DNNs, in fact, are able to achieve interesting performance when addressing very specific problems (sometimes also reaching super-human abilities), but largely fail to properly address multiple tasks. Multi-task learning was actually object of several research in the past years [40]. The typical approach consists in sharing the first one or two hidden layers of the DNN, in order to derive general and more robust feature representations. We believe that a better way to foster cooperation is to exchange higher-level information at multiple levels rather than just sharing part of the architecture.

The network of deep neural network, depicted in Figure 6.1a, follows this philosophy. The proposed paradigm is not based on independent DNNs, but all the systems are organized in a network where a full-communication across the elements arises. Each element solves a different task and optimizes a different cost function. The cooperation with the other modules is realized by exchanging DNN's outputs. In the example of Figure 6.1a, for instance, $DNN1$ is fed by the input features x and by the output of the other DNNs. This network, instead of estimating $P(y_1/x)$, thus estimates posterior probabilities $P(y_1/x, y_2, y_3)$ that are enriched by additional information processed by other DNNs. This information can be regarded as a sort of prior knowledge, that might help to mitigate DNN uncertainty. The contribution of the other systems can be helpful or not. However, instead of planning by hand the system communication, this paradigm leaves the network to freely decide which communication channels are more helpful, minimizing (ideally) human efforts in the architecture design.

Such a communication modality inherently entails a *chicken-and-egg*



(a) A network of DNNs.

(b) Unfolding a network of DNN.

Figure 6.1: An example of network of deep neural networks.

problem, as clearly highlighted in the following recurrent equations:

$$\begin{aligned} y_1 &= f_1(x, y_2, y_3, \theta_1) \\ y_2 &= f_2(x, y_1, y_3, \theta_2) \\ y_3 &= f_3(x, y_1, y_3, \theta_3) \end{aligned}$$

The chicken-and-egg problem is caused by the fact that each DNN is fully connected with the others through a full bidirectional communication channel. To circumvent this issue we suggest to unfold the network of deep neural networks using an approach similar to that adopted for recurrent neural networks. Figure 6.1b shows an example of unfolded network of deep neural networks: at the first level the networks are independent and are fed only by the low-level input features x. A full communication is then established and continuously refined by progressively adding more levels to the architecture.

Algorithm 2 back-propagation through network algorithm

1: Unfold the network of DNN until a communication level L
2: **Forward Pass (from $l = 1$ to $l = N$):**
3: Starting from input x do a forward pass through all the DNNs.
4: **Compute Cost Functions:**.
5: For each DNN i of the level l compute $C_{i,l}$
6: **Gradient back-propagation (from $l = N$ to $l = 1$):**
7: For each DNN i of the level l compute the gradients and back-propagate them through all the connected nodes.
8: The gradient of the parameters $\theta_{i,l}$ will be given by:
$g_{\theta_{i,l}} = \frac{\partial C_{i,l}}{\theta_{i,l}} + \sum_{m=l+1}^{L} \sum_{n=1}^{N} \frac{\partial C_{n,m}}{\theta_{i,l}}$
9: **Parameter Updates:**
10: Given $g_{\theta_{i,l}}$ use an optimizer to compute the new parameters $\theta_{i,l}$

After the unfolding procedure, the overall computational graph can be considered as a single very deep neural network where all the DNNs can be jointly trained. In particular, training can be carried out with the *back-propagation through network*, that is a variation of the standard back-propagation. The proposed algorithm is described in Alg. 2, where x are the input features, L the number of communication levels, N the number of DNNs, g the gradients and θ the DNN parameters. This algorithm is repeated for all the minibatches and iterated for several epochs until convergence. The basic idea is to perform a forward pass, compute the loss functions at the output of each DNN, compute the corresponding gradients, and back-propagate them. The gradient, that is back-propagated through all the connected DNNs, is given by the following equation:

$$g_{\theta_{i,l}} = \underbrace{\frac{\partial C_{i,l}}{\theta_{i,l}}}_{\text{local gradient}} + \underbrace{\sum_{m=l+1}^{L} \sum_{n=1}^{N} \frac{\partial C_{n,m}}{\theta_{i,l}}}_{\text{other contributions}} \tag{6.1}$$

The updates of each DNN not only depend on their local cost, but also on the higher-level losses. In this way the training of the DNNs is in part

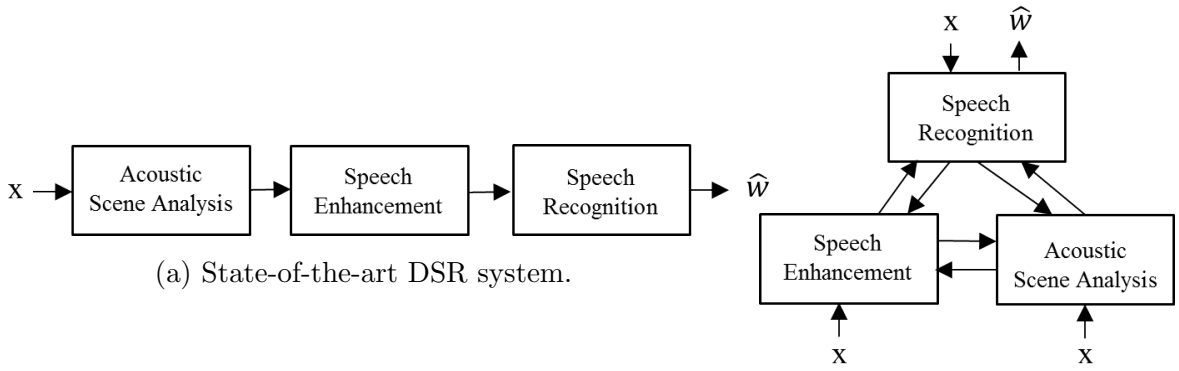
driven by the contribution of the others, that would hopefully be helpful to improve the system performance. As will be discussed in the following sections, the integration of different gradients coming from the higher levels produces a regularization effect, that significantly helps the training of the system.

Note that no direct connections between DNNs solving the same task are considered in the unrolled architecture depicted in Fig. 6.1b (for instance, DNN1 $l = 2$ does not feed DNN1 $l = 3$). Results, not reported here, have shown that such communication channel degrades the overall system performance, since the higher level DNN tends just to copy at the output the information provided by the lower-level DNN.

6.2 Application to DSR

The most advanced state-of-the-art DSR systems are based on rather complex architectures, that are often composed of multiple modules working together [295]. An example is depicted in Figure 6.2a, where a pipeline of acoustic scene analysis (that might be, for instance, a module for environment classification or for acoustic event detection), speech enhancement and speech recognition technologies is shown. Although this pipeline sounds perfectly reasonable and organized in a logical fashion, we believe that current systems suffer from the following prominent limitations:

- **Lack of matching**. Even in modern DSR systems [295], speech enhancement and speech recognition are often developed independently. Moreover, in several cases, the enhancement part is tuned according to metrics which are not directly correlated with the final speech recognition performance. This weakness is due to the historical evolution of DSR technologies: the community working on speech enhancement and speech recognition were, indeed, largely independent for several



(a) State-of-the-art DSR system.

(b) Proposed architecture.

Figure 6.2: Network of DNNs for distant speech recognition.

years. Speech enhancement was mostly based on signal processing techniques, while ASR technology was based on pure machine learning approaches. We strongly believe that deep learning can help break down the wall between these different disciplines. The recent success of DNNs, in fact, has not only largely contributed to a substantial improvement of speech recognition [263, 160, 285, 162, 250, 219, 221], but has also enabled the development of competitive DNN-based speech enhancement solutions [293, 294, 284], making an effective integration between these technologies easier.

- **Lack of communication**. As shown in Figure 6.2a, state-of-the-art system are based on a flat pipeline, based on a unidirectional communication flow. The speech enhancement, for instance, normally helps the speech recognizer, but the output of the latter is not commonly used, in turn, to improve the speech enhancement. We argue that establishing this missing link can nevertheless be very useful, since a hint on the recognized phone sequence might help the speech enhancement in performing its task. This can be particularly helpful under noisy and reverberant conditions: when a significant uncertainty arises, a

continuous interaction between the systems can guide them towards a better decision.

We propose to mitigate the latter issues with the network of deep neural network approach described in the previous section. Figure 6.2b reports the proposed architecture, in which all the basic modules of the DSR system are implemented with DNNs. The lack of matching is mitigated using a joint-training strategy, while the lack of communication is improved through a full communication across DNNs.

To validate our architecture, in the next section we start from a simplified scenario where a flat pipeline of a speech enhancement and speech recognition DNNs is jointly trained with batch normalization. Even though different types of DNNs can be embedded within the network of deep neural networks framework, the studies conducted in this thesis and discussed in the following are focused on standard feed-forward DNNs.

6.3 Batch Normalized Joint Training

Within the DNN framework, one way to achieve a fruitful integration of the various components is joint training. The core idea is to pipeline a speech enhancement and a speech recognition deep neural network and to jointly update their parameters as if they were within a single bigger network. Although joint training for speech recognition is still an under-explored research direction, such a paradigm is progressively gaining more attention and some interesting works in the field have been recently published [186, 283, 81, 45, 242, 177, 116].

In this thesis, we contributed to this line of research by proposing an approach based on batch normalization, that turned out to be very precious to improve the network convergence, since it makes one network less sensitive to changes of the other. Differently to previous works [283, 81],

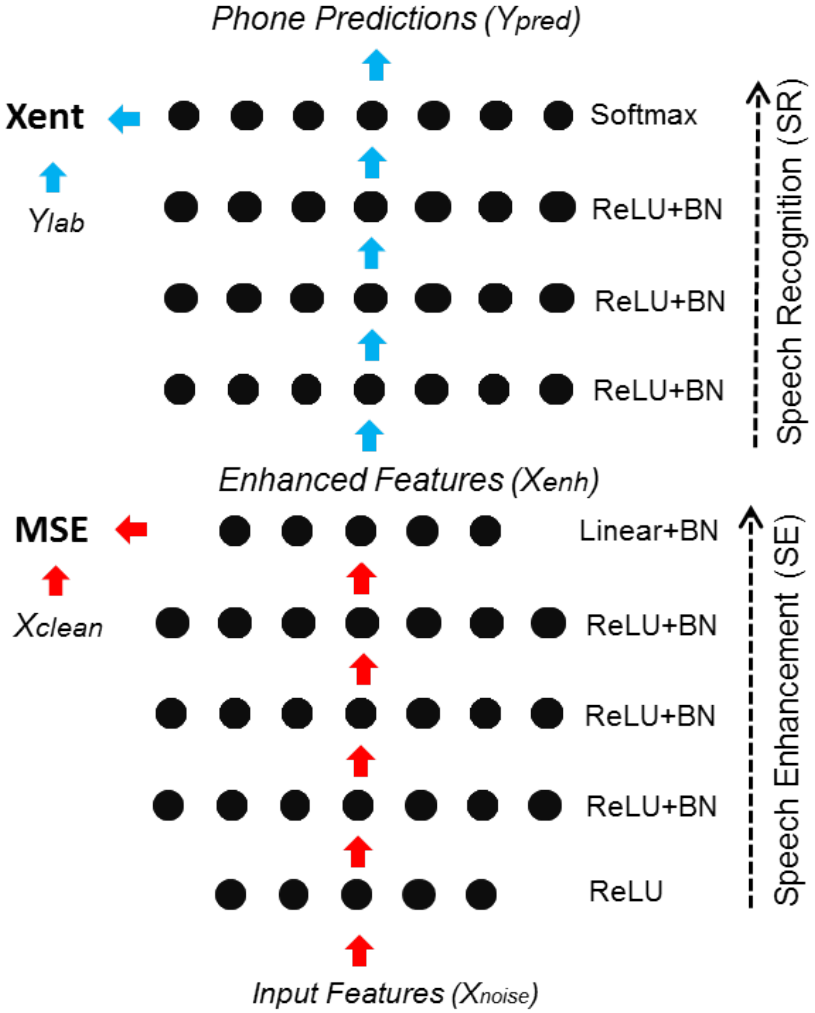


Figure 6.3: The FF-DNN architecture proposed for joint training.

using this approach we were able to jointly train a cascade between speech enhancement and speech recognition DNNs without any pre-training step.

The proposed architecture is depicted in Figure 6.3. A bigger joint DNN is built by concatenating a speech enhancement and a speech recognition MLP. The speech enhancement DNN is fed with the noisy features x_{noise} gathered within a context window and tries to reconstruct at the output the original clean speech (regression task). The speech recognition DNN is fed by the enhanced features x_{enh} estimated at the previous layer and

performs phone predictions y_{pred} at each frame (classification task). The architecture of Figure 6.3 is trained with the algorithm described in Alg. 3.

The basic idea is to perform a forward pass, compute the loss functions at the output of each DNN (mean-squared error for speech enhancement and negative multinomial log-likelihood for speech recognition), compute and weight the corresponding gradients, and back-propagate them. In the joint training framework, the speech recognition gradient is also back-propagated through the speech enhancement DNN. Therefore, at the speech enhancement level, the parameter updates not only depend on the speech enhancement cost function but also on the speech recognition loss, as shown by the following equation:

$$\theta_{SE} \leftarrow \theta_{SE} - lr * (g_{SE} + \lambda g_{SR}) \,. \tag{6.2}$$

Where θ_{SE} are the parameters of the speech enhancement DNN, g_{SE} is the gradient of such parameters computed from the speech enhancement cost function (mean squared error), while g_{SR} is the gradient of θ_{SE} computed from the speech recognition cost function (multinomial log-likelihood). Finally, λ is a hyperparameter for weighting g_{SR} and lr is the learning rate.

The key intuition behind joint training is that since the enhancement process is in part guided by the speech recognition cost function, the front-end would hopefully be able to provide enhanced speech which is more suitable and discriminative for the subsequent speech recognition task. From a machine learning perspective, this solution can also be considered as a way of injecting a useful task-specific prior knowledge into a deep neural network. On the other hand, it is well known that training deep architectures is easier when some hints are given about the targeted function [41]. As shown previously [41], such prior knowledge becomes progressively more precious as the complexity of the problem increases and can thus be

very helpful for a distant speech recognition task. Similarly to the current work, in [41, 233] a task-specific prior knowledge has been injected into an intermediate layer of a DNN for better addressing an image classification problem. In our case, we exploit the prior assumption that to solve our specific problem, it is reasonable to first enhance the features and, only after that, perform the phone classification. Note that this is certainly not the only way of solving the problem, but among all the possible functions able to fit the training data, we force the system to choose from a more restricted subset, potentially making training easier. On the other hand, good prior knowledge is helpful to defeat the curse of dimensionality, and a complementary view is thus to consider the proposed joint training as a regularizer. According to this vision, the weighting parameter λ of Eq. 6.2 can be regarded as a regularization hyperparameter, as will be better discussed in Sec. 6.3.4.

This joint training of the network of DNNs, however, can be complicated by the fact that the output distribution of the DNNs may change substantially during the optimization procedure. Each DNN would have to deal with an input distribution that is non-stationary and unnormalized, possibly causing convergence issues. To mitigate this issue, known as internal covariate shift, we suggest to couple the proposed architecture with batch normalization, that turned out to be crucial for achieving a better performance, to improve DNN convergence, and to avoid any time-consuming pre-training steps. Particular attention should anyway be devoted to the initialization of the γ parameter. Contrary to [123], where it was initialized to unit variance ($\gamma = 1$), in this work we have observed better performance and convergence properties with a smaller variance initialization ($\gamma = 0.1$). A similar outcome has been found in [49], where fewer vanishing gradient problems are empirically observed with small values of γ in the case of recurrent neural networks.

Algorithm 3 Pseudo-code for joint training

DNN initialization
for i in minibatches **do**
    Forward Pass:
    Starting from the input layer do a forward pass
    (with batch normalization) through the networks.
    Compute SE Cost Function:
    $MSE_i = \frac{1}{N}\sum_{n=1}^{N}(x^i_{enh} - x^i_{clean})^2$
    Compute SR Cost Function:
    $NLL_i = -\frac{1}{N}\sum_{n=1}^{N} y^i_{lab} log(y^i_{pred})$
    Backward Pass:
    Compute the grad. g^i_{SE} of MSE_i and backpropagate it.
    Compute the grad. g^i_{SR} of NLL_i and backpropagate it.
    Parameters Updates:
    $\theta^i_{SE} \leftarrow \theta^i_{SE} - lr * (g^i_{SE} + \lambda g^i_{SR})$
    $\theta^i_{SR} \leftarrow \theta^i_{SR} - lr * g^i_{SR}$
Compute NLL on the development dataset
if $NLL_{dev} < NLL^{prev}_{dev}$ **then**
    Train for another epoch (go to 2)
else
    Stop Training

6.3.1 Relation to prior work

Similarly to our work, a joint training framework between speech enhanancement and speech recognition has been explored in [186, 283, 81, 45, 242, 177, 116]. In [283, 81], for instance, the joint training was actually performed as a fine-tuning procedure, which was carried out only after training the two networks independently. This approach can be justified by the fact that jointly train all the DNNs from scratch is actually very challenging with standard deep learning techniques, due to possible convergence issues. However, pre-training significantly slows down the training time, making the overall learning procedure particularly heavy and com-

putational demanding. Another critical aspect of such an approach is that the learning rate adopted in the fine-tuning step has to be properly selected in order to really take advantage of pre-training.

A key difference with previous efforts is that we propose to combine joint training with batch normalization. With this technique we are not only able to significantly improve the performance of the DSR system, but also to perform joint training from scratch, skipping any pre-training phase without convergence issues. Note also that our approach naturally entails a regularization effect: the speech recognition, in fact, will be fed with poor enhanced features at the beginning of training, introducing a sort of noise that could improve DNN generalization.

6.3.2 Joint Training Performance

The experiments reported in the following summarize the main findings reported in [212]. The experimental validation has been conducted using different datasets, acoustic conditions, and tasks. A first set of experiments was conducted with the TIMIT corpus, using a phone-loop task. Training and test has been performed with contaminated versions of this dataset. The impulse responses used for contamination were measured in the living-room of the DIRHA apartment.

The experiments have then be extended to a more realistic WSJ task. Training was performed with a reverberated version of the standard WSJ corpus, while evaluation considered the real recordings of the DIRHA-English-WSJ corpus under both *rev* and *rev&noise* conditions. MFCC features are used for all the experiments. See C.6 for more detailed description of the experimental setup.

As a reference baseline, we first report the close-talking performance achieved with a single DNN. The Phoneme Error Rate (PER%) obtained by decoding the original test sentences of TIMIT is 19.5% (using DNN

System \ *Dataset*	TIMIT	WSJ	WSJ
	Rev	*Rev*	*Rev+Noise*
Single big DNN	31.9	8.1	14.3
SE + clean SR	31.4	8.5	15.7
SE + matched SR	30.1	8.0	13.7
SE + SR joint training	**29.1**	**7.8**	**12.7**

Table 6.1: Performance of the proposed joint training approach compared with other competitive DNN-based systems. Training is performed with WSJ-rev, while test is performed with DIRHA-English-WSJ (set2, real part).

System \ *Dataset*	Without Pre-Training		With Pre-Training	
	no-BN	*with-BN*	*no-BN*	*with-BN*
TIMIT-Rev	34.2	**29.1**	32.6	29.5
WSJ-Rev	9.0	**7.8**	8.8	7.8
WSJ-Rev+Noise	15.7	**12.7**	15.0	12.9

Table 6.2: Analysis of the role played by batch normalization within the proposed joint training framework.

models trained with the original dataset). The Word Error Rate (WER%) obtained by decoding the close-talking DIRHA-English-WSJ sentences is 3.3%. It is worth noting that, under such favorable acoustic conditions, the DNN model leads to a very accurate sentence transcription, especially when coupled with a language model.

In Table 6.1, we reported the performance obtained with distant-talking experiments where the proposed joint training approach is compared with other competitive strategies. In particular, the first line shows the results obtained with a single neural network. The size of the network has been optimized on the development set (4 hidden layers of 1024 neurons for TIMIT, 6 hidden layers of 2048 neurons for WSJ cases). The second line shows the performance obtained when the speech enhancement neural network (4 hidden layers of 2048 neurons for TIMIT, 6 hidden layers of

2048 neurons for WSJ) is trained independently and later coupled with the aforementioned close-talking DNN. These results are particularly critical because, especially in adverse acoustic conditions, the speech enhancement model introduces significant distortions. The close-talking DNN, trained in the usual way, is thus not able to cope with this significant amount of mismatch. To partially recover such a critical mismatch, one approach is to first train the speech enhancement, then pass all the training features through the speech enhancement DNN, and, lastly, train the speech recognition DNN with the dataset processed by the speech enhancement. The third line shows results obtained with such a matched training approach. The last line reports the performance achieved with the proposed joint training approach. Batch normalization is adopted for all the systems considered in Table 6.1.

Although joint training exhibits the best performance in all the cases, it is clear that such a technique is particularly helpful especially when challenging acoustic conditions are met. For instance, a relative improvement of about 8% over the most competitive matched training system is obtained for the WSJ task in noisy and reverberant conditions.

6.3.3 Role of batch normalization

In Table 6.2, the impact of batch normalization on the joint training framework is shown. The first two columns report, respectively, the results obtained with and without batch normalization when no pre-training techniques are employed. The impact of pre-training is studied in the last two columns. The pre-training strategy considered here consists of initializing the two DNNs with the matched training system discussed in the previous section, and performing a fine-tuning phase with a reduced learning rate. The column corresponding to the pre-training without batch normalization represents a system that most closely matches the approaches followed in

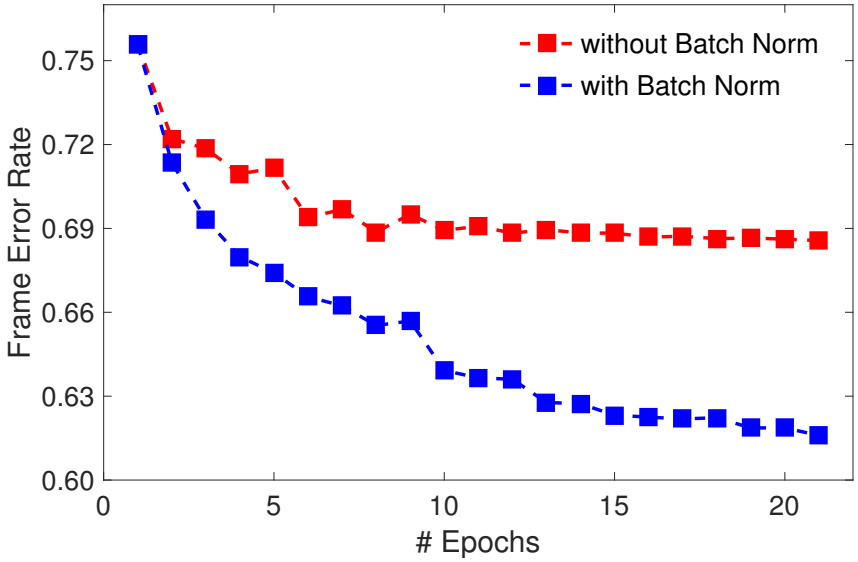


Figure 6.4: Evolution of the test frame error rate across various training epochs with and without batch normalization.

[283, 81].

Table 6.2 clearly shows that batch normalization is particularly helpful. For instance, a relative improvement of about 23% is achieved when batch normalization is adopted for the WSJ task in a noisy and reverberant scenario. The key importance of batch normalization is also highlighted in Figure 6.4, where the evolution during training of the frame-level phone error rate (for the TIMIT-Rev dataset) is reported with and without batch normalization. From the figure it is clear that batch normalization, when applied to the considered deep joint architecture, ensures a faster convergence and a significantly better performance. Moreover, as shown in Table 6.2, batch normalization eliminates the need of DNN pre-training, since similar (or even slightly worse results) are obtained when pre-training and batch normalization are used simultaneously.

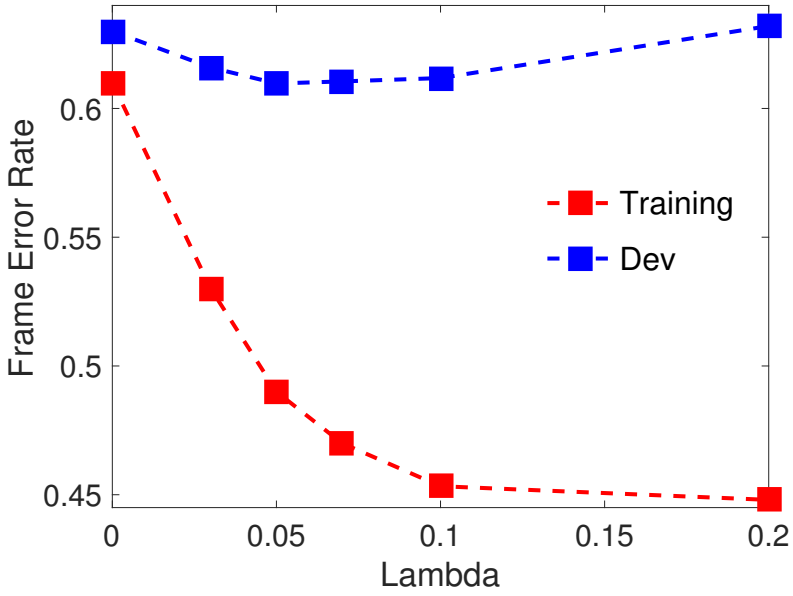


Figure 6.5: Training and development frame error rates obtained on the TIMIT-Rev dataset for different values of λ.

6.3.4 Role of the gradient weighting

In Figure 6.5, the role of the gradient weighting factor λ is highlighted. From the figure one can observe that small values of λ lead to a situation close to underfitting, while higher values of λ cause overfitting. The latter result is somewhat expected since, intuitively, with very large values of λ the speech enhancement information tends to be neglected and training relies on the speech recognition gradient only.

In the present work, we have seen that values of λ ranging from 0.03 to 0.1 provide the best performance. Note that these values are smaller than that considered in [283, 186], where a pure gradient summation ($\lambda = 1$) was adopted. We argue that this result is due to the fact that, as observed in [49], the norm of the gradient decays very slowly when adopting batch normalization with a proper initialization of γ, even after the gradient has passed through many hidden layers. This causes the gradient back-

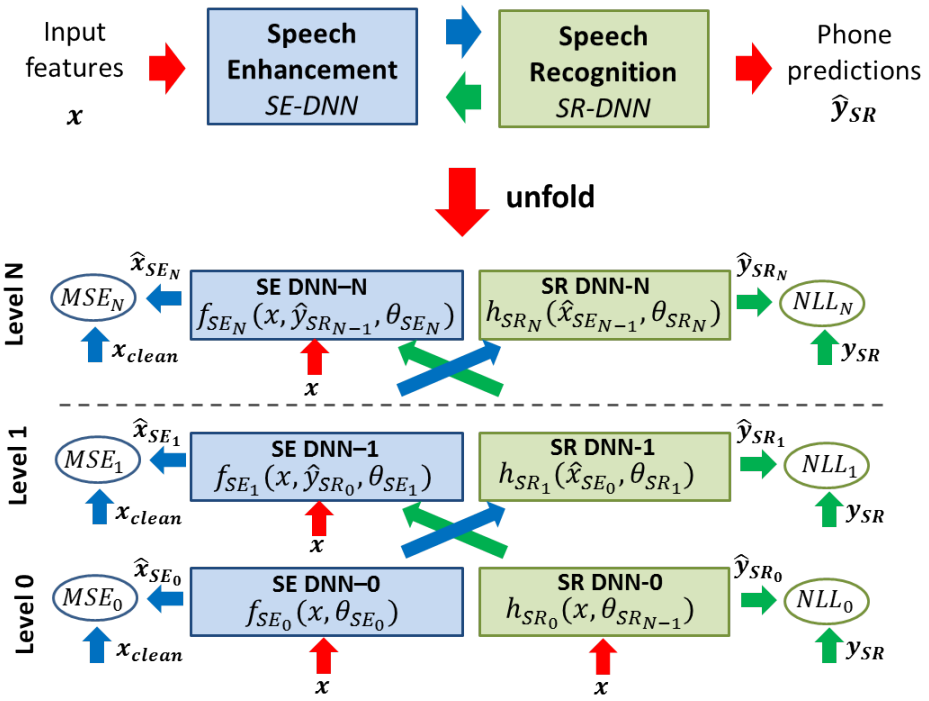


Figure 6.6: The proposed network of deep neural networks for speech enhancement and speech recognition. Each element of the network is a FF-DNN.

propagated through the speech recognition network and into the speech enhancement network to be very large.

6.4 Cooperative Enhancement and Recognition

In this section we evolve the flat joint trained pipeline experimented so far towards a cooperative network of DNNs. As reported in Figure 6.6, this work considers a full communication between the speech enhancement and speech recognition, which are the most critical components of a DSR system. The proposed architecture is unfolded and trained with the backpropagation through network algorithm as described in Sec. 6.1. As discussed in the previous section, all the components are jointly trained with a single learning procedure, and batch normalization is adopted to improve the network convergence.

In this specific application, an effective system communication can be impaired by the high dimensionality of the speech recognizer output $\hat{y}_{SR_\ell}$. This dimensionality derives from the number of considered context-dependent states, which typically ranges from 1000 to 4000 (depending on the phonetic decision tree and on the dataset). To avoid feeding the speech enhancement with such a high dimensional input, we jointly estimate the monophone targets (which are only some dozens). Similarly to [253], this is realized by adding an additional softmax classifier on the top of the last hidden layer of each speech recognition DNN.

6.4.1 Related work

Similarly to this work, an iterative pipeline based on feeding the speech recognition output into a speech enhancement DNN has recently been proposed in [45, 66]. The main difference with our approach is that the latter circumvents the chicken-and-egg problem by simply feeding the speech enhancement with the speech recognition alignments generated at the previous iteration, while our solution faces this issue by adopting the unrolling procedure over different interaction levels previously discussed.

Our paradigm has also some similarities with traditional multi-tasking techniques [202]. The main difference is that the latter are based on sharing some hidden layers across the tasks, while our method relies on exchanging DNN outputs at various interaction levels. Finally, the proposed training algorithm has some aspects in common with the back-propagation through structure originally proposed for the parsing problem [91]. The main difference is that the latter back-propagates the gradient through a tree structure, while the proposed variation back-propagates it on a less constrained network of components. Another difference is that in the original algorithm the same neural network is used across all the levels of the tree, while in this work different types of DNNs (i.e., speech enhancement

and speech recognition) are involved.

6.4.2 Network of DNNs performance

The experiments reported in the following summarize the main achievements emerged in [215]. The experiments were based on the same datasets and with the same experimental setup considered in the previous section. More details can be found in App. C.7.

The proposed network of DNNs approach is compared in Table 6.3 with other competitive systems. The first line reports the results obtained with a single neural network. In this case, only the speech recognition labels are used and the DNN is not forced to perform any speech enhancement task. The second line shows the performance obtained when the single DNN is coupled with a traditional multi-task learning, in which speech enhancement and speech recognition tasks are simultaneously considered, as shown in Figure 6.7. This multi-task architecture shares the first half of the hidden layers across the tasks, while the second half of the architecture is task-dependent. This approach aims to discover (within the shared layers) more general and robust features which can be exploited to better solve both correlated tasks. The third line reports the performance achieved with the joint training approach described in the previous section. In this case a bigger DNN composed of a cascade of a speech enhancement and a speech recognition DNNs is jointly trained by back-propagating the speech recognition gradient also into the speech enhancement DNN. The last line finally shows the performance achieved with the proposed network of deep neural network approach (unfolded up to level 2). To allow a fair comparison, batch normalization is adopted for all the considered systems.

Table 6.3 highlights that the proposed approach significantly outperforms all the single DNN systems. For instance, a relative improvement of about 14% over the single DNN baseline is obtained for the *WSJ rev+noise*

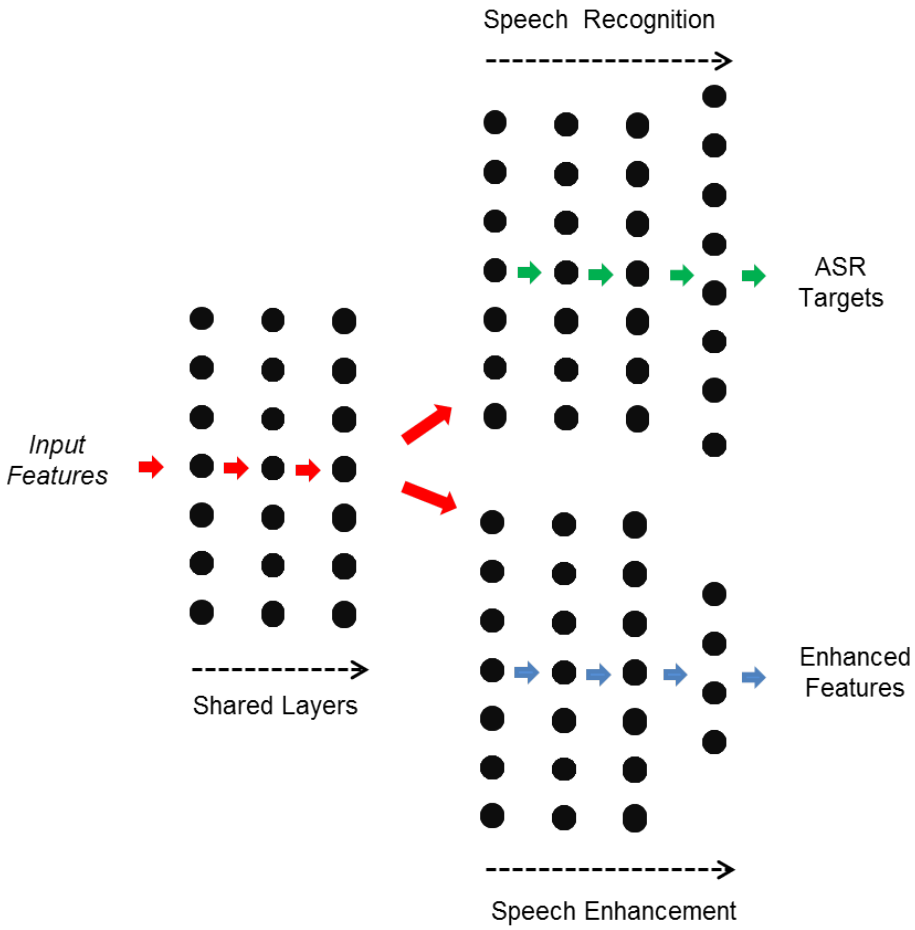


Figure 6.7: Multi-task DNN for speech enhancement and recognition.

Dataset / *System*	TIMIT *rev*	WSJ *rev*	WSJ *rev+noise*
Single DNN	31.9	8.1	14.3
Single DNN +multitask	31.4	8.1	13.8
Joint SE-SR training	29.1	7.8	12.7
Network of DNNs	**28.7**	**7.6**	**12.3**

Table 6.3: Performance of the proposed network of DNN approach compared with other competitive DNN-based systems (PER% for TIMIT, WER% for WSJ). For WSJ experiments, test is performed with DIRHA-English-WSJ (set2, real-part).

case. The network of deep neural networks also outperforms the considered joint training method. This result suggests that the improved cooperation

Dataset	Level 0	Level 1	Level 2
TIMIT rev	31.4	29.1	**28.7**
WSJ rev	8.0	7.7	**7.6**
WSJ rev+noise	14.3	12.7	**12.3**

Table 6.4: Performance of the proposed network of DNN achieved at various levels of the architecture.

between the networks achieved with our full communication scheme can overtake the standard DSR pipeline based on a partial and unidirectional information flow (which is still considered in the context of joint training approaches).

Table 6.4 shows the results obtained by decoding the speech recognition output at the various levels of the proposed architecture (denoted as $\hat{y}_{SR_0}$, $\hat{y}_{SR_1}$, $\hat{y}_{SR_2}$ in Figure 6.6). One can note that the performance become progressively better as the level of the network of DNNs increases till level 2. As expected, the first level speech recognizer (*SR DNN-0*) performs similarly to the single DNN baseline. The second level (*SR DNN-1*) is based on a simple cascade between a speech enhancement and a speech recognition DNNs, and thus provides results similar to that obtained with standard joint training. The third level (*SR DNN-2*) achieves the best performance, confirming that the progressive interaction of the DNNs involved in the DSR process helps in improving the system performance. No additional benefits have been observed for the considered tasks by adding more than 3 levels. The number of communication levels, however, can be regarded as an hyperparameter of the architecture and its best value might be task-dependent. In general, one should anyway expect a diminishing return when increasing N.

6.5 Summary and Future Challenges

This Chapter discussed the network of deep neural networks paradigm and its application to DSR. The experimental part started with a simplified scenario, where a flat pipeline between speech enhancement and speech recognition was jointly trained with batch normalization. From these experiments emerged the great importance of this learning modality, that turned out to improve matching between the various components. Our efforts were then focused on a network of DNNs between speech recognition and speech enhancement. The experiments have shown that cooperation and full communication across these modules is of key importance for counteracting the uncertainty originated by noise and reverberation.

The proposed paradigm, however, is not yet fully mature and the real potential of this framework, according to us, is still far from being reached. Several efforts will be devoted to derive proper architectures able to foster the cooperation across the various DNNs. For instance, some preliminary results, reported in [215], show that the adoption of Residual Neural Networks (ResNet) is a very promising architectural variation, which can significantly improve gradient back-propagation.

Moreover, inspired by the gating mechanisms used in the context of recurrent networks, we are studying the adoption of learnable communication gates to manage the information flow through the unfolded computation graph. The network of DNNs proposed in this thesis was limited to speech enhancement and speech recognition. In the future we will add other components, possibly including systems for acoustic scene analysis, such as environmental classification, acoustic event detection, unsupervised estimation of the reverberation time, speaker identification and speaker verification.

We will also explore the extension of this paradigm to RNNs. Although

this is rather natural and straightforward, a possible issue is the considerable memory required for unfolding the computational graph over both time and communication levels, making an efficient multi-gpu implementation strictly necessary.

Finally, we would like to highlight that the proposed framework is a general paradigm that can be used in other fields. Cooperative DNNs can be considered, for instance, in application such as robotics, where multiple components have to properly work together to achieve a common global goal.

Chapter 7

Conclusion

The results reported in this PhD thesis summarize the main efforts undertaken over the last four years. During that time, the research community has experienced a revolution, thanks to the popularization and maturation of deep learning techniques. When this doctorate began in November 2013, the actual impact of this technology was still not clear. Today, after only few years of intense research, deep learning is a consolidated state-of-the-art technology for ASR and we can consider this paradigm as a major breakthrough, that can be compared, in term of impact to the research community, only to the widespread diffusion of HMMs, that were proposed more than thirty years ago.

Since the beginning of this PhD, we were totally convinced of the great potential of this framework, especially to face challenging acoustic conditions characterized by significant levels of noise and reverberation. In the latter scenarios, previous HMM-GMM technology was, indeed, very far from achieving a satisfactory robustness, and we were confident that the significant performance gain originally observed with deep learning in the context of standard close-talking ASR could also be extended to DSR. This thesis, and the related papers, represent one of the first attempts to revise standard deep learning algorithms, architectures ,and techniques for

better addressing the specific problem of distant speech recognition. The dominant approach consisted, in most of the cases, of developing proper solutions for close-talking ASR and later inherit them for DSR. Conversely, we directly addressed the DSR application, since we believe that the peculiarities and challenges arising with distant speech deserve specific studies and methodologies.

In particular, we focused our efforts on the main flaws and weaknesses of DSR technology. First, we concentrated on realistic data contamination. Data, in fact, are a crucial factor for the success of deep learning and a proper methodology for realistic data simulation in reverberant environment has been proposed and extensively validated. The realistic simulations have been used for generating high-quality multi-microphone simulated datasets, that have been released at international level. Moreover, some methodologies were proposed to better exploit contaminated data for DNN training.

Another contribution of the thesis was the development of techniques for managing large time contexts. Our research has initially considered feed-forward neural networks, studying the role of asymmetric context windows to counteract the harmful effects of reverberation. We then focused on RNNs, proposing a novel architecture called Light GRU, that is able to better process time contexts, improve system performance, and significantly speed up DNN training.

Inspired by the idea that uncertainty can also be counteracted with cooperation, we finally developed a novel deep learning paradigm called network of deep neural networks. The proposed framework is based on a full communication and interaction across the various DNN modules composing a DSR system and turned out to be very effective to improve the system performance.

The different results, presented in this dissertation, support the posi-

tive accomplishment of our main goal: contributing to reduce the gap between the possibility offered by current technology and user's expectations. In particular, the choice of specifically studying deep learning for distant speech recognition was not only a forward-looking view, but allowed us to establish a niche in which our research was internationally appreciated.

Despite our contribution and the remarkable efforts of the research community working in the field, the road towards human or super-human distant speech recognition performance is still long. The future development of ASR and DSR technology will be closely influenced by the advances in deep learning and, more in general, by the evolution of AI. As for many other fields, we believe that a crucial role will be played by unsupervised learning. The vast majority of speech data, in fact, are available without a corresponding transcription and the development of techniques for better exploiting unlabelled data will be of crucial importance. With this regard, the adoption of generative adversarial training in the context of speech recognition represents a promising research direction.

Moreover, current deep learning models are discovering representations that are still working on low-levels of abstraction, thus focusing on rather superficial aspects of the world. It will be necessary to develop new algorithms and architectures that are able to better disentangle the underlying factors of variability. For DSR, proper solutions for deeply understanding and modeling the actual complexity of acoustic environments will have a major impact.

We also believe that long-life learning will play an important role in the next generation of speech recognizers. The study of suitable never-ending learning paradigms, that might operate in a distributed crowd-sourcing scenario, represents a promising direction towards human or super-human ASR. Reinforcement learning is also a very interesting and under-explored research direction for speech recognition. The study of automatic strate-

gies for exploiting user feed-backs will be crucial for improving the system performance.

We finally think that a major accomplishment, which can potentially have an important impact on several applications, would be the development of a unified learning framework, where supervised, unsupervised, reinforced and long-life learning are jointly exploited within a network of deep neural network paradigm.

Bibliography

[1] O. Abdel-Hamid, A. R. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu. Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(10):1533–1545, Oct 2014.

[2] M. Adda-Decker and L. Lamel. *The Use of Lexica in Automatic Speech Recognition*, pages 235–266. Springer Netherlands, 2000.

[3] J. B. Allen and D. A. Berkley. Image method for efficiently simulating smallroom acoustics. In *Journal of Acoustic Society of America*, pages 2425–2428, 1979.

[4] D. Amodei et al. Deep speech 2: End-to-end speech recognition in english and mandarin. abs/1512.02595, 2015.

[5] B. Angelini, F. Brugnara, D. Falavigna, D. Giuliani, R. Gretter, and M. Omologo. Speaker independent continuous speech recognition using an acoustic-phonetic Italian corpus. In *Proc. of ICSLP*, pages 1391–1394, 1994.

[6] L. Armani, M Matassoni, M. Omologo, and P. Svaizer. Use of a CSP-based voice activity detector for distant-talking ASR. In *Proc. of Eurospeech 2003*, pages 501–504, 2003.

[7] M. El Ayadi, M. S. Kamel, and F. Karray. Survey on speech emotion recognition: Features, classification schemes, and databases. *Pattern Recognition*, 44(3):572 – 587, 2011.

[8] M. Baccouche, B. Besset, P. Collen, and O. Le Blouch. Deep learning of split temporal context for automatic speech recognition. In *Proc. of ICASSP*, pages 5422–5426, 2014.

[9] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio. End-to-End Attention-based Large Vocabulary Speech Recognition. In *Proc. of ICASSP*, pages 4945–4949, 2016.

[10] J. Barker, R. Marxer, E. Vincent, and S. Watanabe. The third CHiME Speech Separation and Recognition Challenge: Dataset, task and baselines. In *Proc. of ASRU*, pages 504–511, 2015.

[11] J. Barker, E. Vincent, N. Ma, H. Christensen, and P. Green. The PASCAL CHiME speech separation and recognition challenge. *Computer Speech and Language*, 27(3):621–633, 2013.

[12] L.E. Baum. An inequality and associated maximization technique in statistical estimation for probabilistic functions of markov processes. *Inequalities*, 3:1–8, 1972.

[13] J. Benesty, J. Chen, and Y. Huang. On the Importance of the Pearson Correlation Coefficient in Noise Reduction. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(4):757–765, May 2008.

[14] J. Benesty, S. Makino, and J. Chen. *Speech Enhancement*. Springer, 2005.

[15] Y. Bengio. Learning Deep Architectures for AI. *Fundamental Trends in Machine Learning*, 2(1):1–127, 2009.

[16] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Transaction Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

[17] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin. A neural probabilistic language model. *Journal of Machine Learning Resources*, 3:1137–1155, 2003.

[18] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *Proc. of ICML*, pages 41–48, 2009.

[19] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transaction on Neural Networks*, 5(2):157–166, March 1994.

[20] J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13:281–305, 2012.

[21] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[22] C. M. Bishop. Regularization and complexity control in feed-forward networks. In *Proc. of ICANN*, pages 141–148, 1995.

[23] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, 2006.

[24] J. Bitzer and K.U. Simmer. Superdirective microphone arrays. In *Microphone Arrays*, pages 19–38. Springer Berlin Heidelberg, 2001.

[25] M. Bonnefond, S. Kastner, and O. Jensen. Communication between brain areas based on nested oscillations. *eNeuro*, 2017.

[26] N. Bostrom. *Superintelligence: Paths, Dangers, Strategies*. Oxford University Press, 1st edition, 2014.

[27] H. Bourlard and N. Morgan. Continuous speech recognition by connectionist statistical methods. *IEEE Transactions on Neural Networks*, 4(6):893–909, 1993.

[28] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[29] D. W. M. Brandstein. *Microphone Arrays: Signal Processing Techniques and Applications*. Springer, 2001.

[30] A. Bregman. *Auditory scene analysis: The perceptual organization of sound. Cambridge*. MIT Press, 1990.

[31] L. Breiman. Bagging predictors. In *Journal of Machine Learning*, pages 123–140, 1996.

[32] A. Brutti and M. Matassoni. On the use of Early-to-Late Reverberation Ratio for ASR in reverberant environments. In *Proc. of ICASSP*, pages 4670–4675, 2014.

[33] A. Brutti and F. Nesta. Tracking of multidimensional tdoa for multiple sources with distributed microphone pairs. *Computer Speech and Language*, 27(3):660–682, 2013.

[34] A. Brutti, M. Omologo, and P. Svaizer. Oriented global coherence field for the estimation of the head orientation in smart rooms equipped with distributed microphone arrays. In *Proc. of Eurospeech*, pages 2337–2340, 2005.

[35] A. Brutti, M. Omologo, and P. Svaizer. Multiple source localization based on acoustic map de-emphasis. *EURASIP Journal on Audio Speech Music Processing*, 2010:11–17, 2010.

[36] A. Brutti, M. Ravanelli, and M. Omologo. SASLODOM: Speech Activity detection and Speaker LOcalization in DOMestic environments. In *Proc. of EVALITA*, 2014.

[37] A. Brutti, M. Ravanelli, P. Svaizer, and M. Omologo. A speech event detection/localization task for multiroom environments. In *Proc. of HSCMA 2014*, pages 157–161.

[38] M. Büchler, S. Allegro, S. Launer, and N. Dillier. Sound classification in hearing aids inspired by auditory scene analysis. *EURASIP Journal on Advances in Signal Processing*, 2005(18):337–845, Nov 2005.

[39] S. Rota Bulò, L. Porzi, and P. Kontschieder. Dropout distillation. In *Proc. of ICML*, pages 99–107, 2016.

[40] R. Caruana. Multitask learning. *Journal of Machine Learning*, 28(1):41–75, July 1997.

[41] Ç. Gülçehre and Y. Bengio. Knowledge matters: Importance of prior information for optimization. *Journal of Machine Learning Research*, 17(8):1–32, 2016.

[42] B. Chen, S. Chang, and S. Sivadas. Learning discriminative temporal patterns in speech: Development of novel traps-like classifiers. In *Proc. of Eurospeech*, pages 429–432, 2001.

[43] G. Chen, C. Parada, and G. Heigold. Small-footprint keyword spotting using deep neural networks. In *Proc. of ICASSP*, pages 4087–4091, 2014.

[44] T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, and Z. Zhang. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. *CoRR*, abs/1512.01274, 2015.

[45] Z. Chen, S. Watanabe, H. Erdogan, and J. Hershey. Speech enhancement and recognition using multi-task learning of long short-term memory recurrent neural networks. In *Proc. of Interspeech*, pages 3274–3278, 2015.

[46] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches. In *Proc. of SSST*, 2014.

[47] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *Proc. of NIPS*, 2014.

[48] R. Collobert, S. Bengio, and J. Marithoz. Torch: A modular machine learning software library, 2002.

[49] T. Cooijmans, N. Ballas, C. Laurent, Ç. Gülçehre, and A. Courville. Recurrent batch normalization. *arXiv preprint arXiv:1603.09025*, 2016.

[50] L. Couvreur, C. Couvreur, and C. Ris. A corpus-based approach for robust ASR in reverberant environments. In *Proc. of Interspeech 2000*, pages 397–400.

[51] L. Cristoforetti, M. Ravanelli, M. Omologo, A. Sosi, A. Abad, M. Hagmueller, and P. Maragos. The DIRHA simulated corpus. In *Proc. of LREC*, pages 2629–2634, 2014.

[52] X. Cui, V. Goel, and B. Kingsbury. Data augmentation for deep neural network acoustic modeling. In *Proc. of ICASSP*, pages 5582–5586, 2014.

[53] S. Cumani, N. Brummer, L. Burget, and P. Laface. Fast discriminative speaker verification in the i-vector space. In *Proc of ICASSP*, pages 4852–4855, 2011.

[54] S. Cumani and P. Laface. e-vectors: JFA and i-vectors revisited. In *Proc. of ICASSP*, pages 5435–5439, 2017.

[55] G. E. Dahl, D. Yu, L. Deng, and A. Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):30–42, 2012.

[56] Y. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Proc. of NIPS*, pages 2933–2941. 2014.

[57] P. Davis, S. Mermelstein. Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(3):621–633, 1980.

[58] R. DeMori. *Spoken Dialogues with Computers*. Academic Press, London, 1998. Chapter 2.

[59] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

[60] D. Dimitriadis, A. Metallinou, I. Konstantinou, G. Goumas, P. Maragos, and N. Koziris. GRIDNEWS: A distributed automatic Greek broadcast transcription system. In *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2009.

[61] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

[62] J. Eaton, N. D. Gaubitch, A. H. Moore, and P. A. Naylor. Estimation of room acoustic parameters: The ace challenge. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(10):1681–1693, 2016.

[63] A. E. Eiben and J. E. Smith. *Introduction to Evolutionary Computing*. Springer-Verlag, 2003.

[64] E. Eide and Herbert Gish. A parametric approach to vocal tract length normalization. In *Proc. of ICASSP*, pages 346–348, 1996.

[65] Y. Ephraim and D. Malah. Speech Enhancement Using a Minimum Mean-Square Error Short-Time Spectral Amplitude Estimator. *IEEE Transactions on Audio, Speech, and Language Processing*, 32(6):1109–1121, 1984.

[66] H. Erdogan, J. R. Hershey, S. Watanabe, and J. Le Roux. Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks. In *Proc. of ICASSP*, pages 708–712, 2015.

[67] D. Silver et al. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, January 2016.

[68] M. Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[69] F. Eyben, F. Weninger, S. Squartini, and B. Schuller. Real-life voice activity detection with LSTM Recurrent Neural Networks and an application to Hollywood movies. In *Proc. of ICASSP*, pages 483–487, 2013.

[70] D. Falavigna, M. Matassoni, S. Jalalvand, M. Negri, and M. Turchi. DNN adaptation by automatic quality estimation of ASR hypotheses. *Computer Speech and Language*, 46:585–604, 2017.

[71] A. Farina. Simultaneous measurement of impulse response and distortion with a swept-sine technique. In *Proc. of the 108th AES Convention*, pages 18–22, 2000.

[72] M. Federico, L. Bentivogli, M. Paul, and S. Stüker. Overview of the IWSLT 2011 evaluation campaign. In *Proc. of IWSLT*, 2011.

[73] X. Feng, Y. Zhang, and J. R. Glass. Speech feature denoising and dereverberation via deep autoencoders for noisy reverberant speech recognition. *Proc. of ICASSP*, pages 1759–1763, 2014.

[74] G. Ferroni, R. Bonfigli, E. Principi, S. Squartini, and F. Piazza. A deep neural network approach for voice activity detection in multi-room domestic scenarios. In *Proc. of IJCNN*, pages 1–8, 2015.

[75] J. G. Fiscus. A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER). In *Proc. of ASRU*, pages 347–354, 1997.

[76] C. Guerrero Flores, G. Tryfou, and M. Omologo. Cepstral distance based channel selection for distant speech recognition. *Computer Speech Language*, 47(Supplement C):314 – 332, 2018.

[77] J. W. Forgie and C. D. Forgie. Results obtained from a vowel recognition computer program. *Journal of Acoustic Society of America*, 31(11):1480–1489, 1959.

[78] S. Furui. 50 years of progress in speech and speaker recognition research. *ECTI Transactions on Computer and Information Technology*, 1(2):64–74, 2005.

[79] Y. Gal and Z. Ghahramani. A theoretically grounded application of dropout in recurrent neural networks. In *Proc. of NIPS*, 2016.

[80] M. Gales and S. Young. The application of hidden markov models in speech recognition. *Foundations and Trends in Signal Processing*, 1(3):195–304, January 2007.

[81] T. Gao, J. Du, L. R. Dai, and C. H. Lee. Joint training of front-end and back-end deep neural networks for robust speech recognition. In *Proc. of ICASSP*, pages 4375–4379, 2015.

[82] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren. DARPA TIMIT Acoustic Phonetic Continuous Speech Corpus CDROM, 1993.

[83] J. F. Gemmeke, D. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *Proc. of ICASSP*, 2017.

[84] O. Gencoglu, T. Virtanen, and H. Huttunen. Recognition of Acoustic Events Using Deep Neural Networks. In *Proc. of EUSIPCO*, 2014.

[85] F. A. Gers and J. Schmidhuber. Recurrent nets that time and count. In *Proc. of the INNS-ENNS*, pages 189–194 vol.3, 2000.

[86] P. Ghahremani, B. BabaAli, D. Povey, K. Riedhammer, J. Trmal, and S. Khudanpur. A pitch extraction algorithm tuned for automatic speech recognition. In *Proc. of ICASSP*, pages 2494–2498, 2014.

[87] D. Gillick, L. Gillick, and S. Wegmann. Don't multiply lightly: Quantifying problems with the acoustic model assumptions in speech recognition. In *Proc. of ASRU*, 2011.

[88] L. Gillick and S. Cox. Some statistical issues in the comparison of speech recognition algorithms. In *Proc. of ICASSP*, pages 532–535, 1989.

[89] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proc. of AISTATS*, pages 249–256, 2010.

[90] J. J. Godfrey, E. C. Holliman, and J. McDaniel. Switchboard: telephone speech corpus for research and development. In *Proc. of ICASSP*, pages 517–520, 1992.

[91] C. Goller and A. Kuchler. Learning task-dependent distributed representations by backpropagation through structure. In *Proc. of ICNN*, pages 347–352, 1996.

[92] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[93] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Proc. of NIPS*, pages 2672–2680, 2014.

[94] I. J. Goodfellow, D. Warde-farley, M. Mirza, A. Courville, and Y. Bengio. Maxout networks. In *Proc. of ICML*, 2013.

[95] A. Graves. Practical variational inference for neural networks. In *Advances in Neural Information Processing Systems*, pages 2348–2356, 2011.

[96] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proc. of ICML*, pages 369–376, 2006.

[97] A. Graves, N. Jaitly, and A.R. Mohamed. Hybrid speech recognition with Deep Bidirectional LSTM. In *Proc. of ASRU*, 2013.

[98] A. Graves, A. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *Proc. of ICASSP*, pages 6645–6649, 2013.

[99] A. Graves, G. Wayne, M. Reynolds, T. Harley, I. Danihelka, A. Grabska-Barwiska, S. G. Colmenarejo, E. Grefenstette, T. Ramalho, J. Agapiou, A. P. Badia, K. M. Hermann, Y. Zwols, G. Ostrovski, A. Cain, H. King, C. Summerfield, P. Blunsom, K. Kavukcuoglu, and D. Hassabis. *Nature*, 538(7626):471–476, 2016.

[100] K. Greff, R. K. Srivastava, J. Koutnk, B. R. Steunebrink, and J. Schmidhuber. LSTM: A Search Space Odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, PP(99):1–11, 2016.

[101] R. Gretter. Euronews: a multilingual speech corpus for ASR. In *Proc. of LREC*, 2014.

[102] F. Grézl and M. Karafiát. Hierarchical Neural Net Architectures for Feature Extraction in ASR. In *Proc. of Interspeech*, pages 1201–1204, 2010.

[103] F. Grézl, M. Karafiát, and L. Brurget. Investigation into Bottle-Neck Features for Meeting Speech Recognition. In *Proc. of Interspeech*, pages 2947–2950, 2009.

[104] F. Grézl, M. Karafiát, S. Kontár, and J. Černocký. Probabilistic and bottle-neck features. In *Proc. of ICASSP*, pages 757–760, 2007.

[105] C. Guerrero, G. Tryfou, and M. Omologo. Channel selection for distant speech recognition exploiting cepstral distance. In *Proc. of Interspeech*, pages 1986–1990, 2016.

[106] A. Guozhong. The effects of adding noise during backpropagation training on a generalization performance. *Neural Computation*, 8(3):643–674, 1996.

[107] E. Habets. Room Impulse Response Generator. In *Technical Report*, 2010.

[108] T. Haderlein, E. Nöth, W. Herbordt, W. Kellermann, and H. Niemann. Using Artificially Reverberated Training Data in Distant-Talking ASR. volume 3658 of *Lecture Notes in Computer Science*, pages 226–233. Springer, 2005.

[109] E. Hänsler and G. Schmidt. *Speech and Audio Processing in Adverse Environments*. Springer, 2008.

[110] S. Haykin. *Blind Deconvolution*. Prentice-Hall, 1994.

[111] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proc. of ICCV*, pages 1026–1034, 2015.

[112] K. He, X. Zhang, S. Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proc. of CVPR*, 2016.

[113] H. Hermansky. Perceptual linear predictive (PLP) analysis of speech. *Journal of Acoustic Society of America*, 28(3):621–633, 1990.

[114] H. Hermansky, D. Ellis, and S. Sharma. Tandem connectionist feature extraction for conventional HMM systems. In *Proc. of ICASSP*, pages 1635–1638, 2000.

[115] H. Hermansky and S. Sharma. TRAPS-Classifiers of temporal patterns. In *Proc. of ICSLP*, pages 1003–1006, 1998.

[116] T. Higuchi, T. Yoshioka, and T. Nakatani. Optimization of speech enhancement front-end with speech recognition-level criterion. In *Proc. of Interspeech*, 2016.

[117] G.E. Hinton, Osindero S., and Teh Y. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(1):1527–1554,, 2006.

[118] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, November 1997.

[119] T. Hori, S. Araki, T. Yoshioka, M. Fujimoto, S. Watanabe, T. Oba, A. Ogawa, K. Otsuka, D. Mikami, K. Kinoshita, T. Nakatani, A. Nakamura, and J. Yamato. Low-latency real-time meeting recognition and understanding using distant microphones and omni-directional camera. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(2):499–513, 2012.

[120] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Journal of Neural Networks*, 2(5):359–366, 1989.

[121] X. Huang, A. Acero, and H.W. Hon. *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. Prentice Hall PTR, 2001.

[122] T. Hughes and K. Mierle. Recurrent neural networks for voice activity detection. In *Proc. of ICASSP*, pages 7378–7382, 2013.

[123] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. of ICML*, pages 448–456, 2015.

[124] K. Irie, Z. Túske, T. Alkhouli, R. Schlúter, and H. Ney. LSTM, GRU, Highway and a Bit of Attention: An Empirical Overview for

Language Modeling in Speech Recognition. In *Proc. of Interspeech*, 2016.

[125] F. Itakura and S. Saito. A statistical method for estimation of speech spectral density and formant frequencies. *electronics and Communications in Japan*, 53:36–43, 1970.

[126] M. G. Christensen J. Benesty, J. R. Jensen and J. Chen. *Speech Enhancement–A Signal Subspace Perspective*. Academic Press, 2015.

[127] K. Jarrett, K. Kavukcuoglu, M.A. Ranzato, and Y. LeCun. What is the best multi-stage architecture for object recognition? In *Proc. of ICCV*, pages 2146–2153, 2009.

[128] M. Jeub, C. Nelke, C. Beaugeant, and P. Vary. Blind estimation of the coherent-to-diffuse energy ratio from noisy speech signals. In *Proc. of EUSIPCO*, pages 1347–1351, 2011.

[129] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proc. of the ACM-ICM*, pages 675–678, 2014.

[130] K. Jim, C. L. Giles, and B. G. Horne. Effects of noise on convergence and generalization in recurrent networks. In *Proc. of NIPS*, 1995.

[131] R. Jozefowicz, W. Zaremba, and I. Sutskever. An empirical exploration of recurrent network architectures. In *Proc. of ICML*, pages 2342–2350, 2015.

[132] R. Biddulph K. H. Davis and S. Balashek. Automatic recognition of spoken digits. *Journal of Acoustic Society of America*, 24(6):627–642, 1952.

[133] M. Kajala and M. Hamalainen. Filter-and-sum beamformer with adjustable filter characteristics. In *Proc. of ICASSP*, pages 2917–2920, 2001.

[134] K. Kawaguchi. Deep learning without poor local minima. In *Proc. of NIPS*, pages 586–594. 2016.

[135] W. Kellermann. *Beamforming for Speech and Audio Signals*. in Hand-Book of Signal Processing in Acoustics, Springer, 2008.

[136] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and Pierre Dumouchel. A study of inter-speaker variability in speaker verification. *IEEE Transactions on Audio, Speech and Language Processing*, 16(5):980–988, 2008.

[137] D.P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proc. of ICLR*, 2015.

[138] K. Kinoshita, M. Delcroix, T. Yoshioka, T. Nakatani, E. Habets, R. Haeb-Umbach, V. Leutnant, A. Sehr, W. Kellermann, R. Maas, S. Gannot, and B. Raj. The reverb challenge: A Common Evaluation Framework for Dereverberation and Recognition of Reverberant Speech. In *Proc. of WASPAA*, pages 1–4, 2013.

[139] M. Kleinschmidt. Methods for capturing spectro-temporal modulations in automatic speech recognition. *Acta Acustica united with Acustica*, 88(3):416–422, 2002.

[140] C. H. Knapp and G. C. Carter. The generalized correlation method for estimation of time delay. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 24(4):320–327, 1976.

[141] R. Kneser and H. Ney. Improved backing-off for m-gram language modeling. In *Proc. of ICASSP*, pages 181–184, 1995.

[142] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur. Audio augmentation for speech recognition. In *Proc. of Interspeech*, pages 3586–3589, 2015.

[143] M. Kolbcek, Z. H. Tan, and J. Jensen. Speech enhancement using long short-term memory based recurrent neural networks for noise robust speaker verification. In *Proc. of SLT*, pages 305–311, 2016.

[144] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009.

[145] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Neural Computation*, 2012.

[146] A. Kumar and D. Florêncio. Speech enhancement in multiple-noise conditions using deep neural networks. In *Proc. of Interspeech*, pages 3738–3742, 2016.

[147] H. Kuttruff. *Room acoustic*. Spon Press, 5 edition, 2009.

[148] C. Laurent, G. Pereyra, P. Brakel, Y. Zhang, and Y. Bengio. Batch normalized recurrent neural networks. In *Proc. of ICASSP*, pages 2657–2661, 2016.

[149] Q.V. Le, N. Jaitly, and G.E. Hinton. A simple way to initialize recurrent networks of rectified linear units. *CoRR*, abs/1504.00941, 2015.

[150] B. Lecouteux, M. Vacher, and F. Portet. Distant Speech Recognition in a Smart Home: Comparison of Several Multisource ASRs in Realistic Conditions. In *Proc. of Interspeech*, pages 2273–2276, 2013.

[151] Y. LeCun and Y. Bengio. The handbook of brain theory and neural networks. chapter Convolutional Networks for Images, Speech, and Time Series, pages 255–258.

[152] Y. LeCun, L. Bottou, G. Orr, and K. Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–50. Springer Berlin Heidelberg, 1998.

[153] B. Lee and D. Ellis. Noise robust pitch tracking using subband autocorrelation classification (SAcC). In *Proc. of Interspeech*, 2012.

[154] E. Lehmann and A. Johansson. Prediction of energy decay in room impulse responses simulated with an image-source model. In *Journal of Acoustic Society of America*, volume 124(1), pages 269–277, 2008.

[155] X. Lei, A. Senior, A. Gruenstein, and J. Sorensen. Accurate and compact large vocabulary speech recognition on mobile devices. In *Proc. of Interspeech*, pages 662–665, 2013.

[156] G. W. Leibniz. Nova methodus pro maximis et minimis, itemque tangentibus, quae nec fractas, nec irrationales quantitates moratur, et singulare pro illis calculi genus. *Acta Eruditorum*, pages 467–473, 1684.

[157] B. Li and K. C. Sim. Modeling long temporal contexts for robust dnn-based speech recognition. In *Proc. of Interspeech*, pages 353–357, 2014.

[158] Z. C. Lipton. A critical review of recurrent neural networks for sequence learning. *CoRR*, abs/1506.00019, 2015.

[159] D. Liu, P. Smaragdis, and M. Kim. Experiments on deep learning for speech denoising. In *Proc. of Interspeech*, pages 2685–2689, 2014.

[160] Y. Liu, P. Zhang, and T. Hain. Using neural network front-ends on far field multiple microphones based speech recognition. In *Proc. of ICASSP*, pages 5542–5546, 2014.

[161] L. Lu and S. Renals. Small-footprint Deep Neural Networks with Highway Connections for Speech Recognition. In *Proc. of Interspeech*, pages 12–16, 2016.

[162] S. Sakai M. Mimura and T. Kawahara. Reverberant speech recognition combining deep neural networks and deep autoencoders. In *Proc. of IEEE REVERB Workshop*, 2014.

[163] A. Maas, Q. V. Le, T. M. O'Neil, O. Vinyals, P. Nguyen, and A. Y. Ng. Recurrent neural networks for noise reduction in robust asr. In *Proc. of Interspeech*, 2012.

[164] A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. of ICML*, 2013.

[165] H. K. Maganti and M. Matassoni. Auditory processing-based features for improving speech recognition in adverse acoustic conditions. *EURASIP Journal on Audio, Speech, and Music Processing*, 2014(1), May 2014.

[166] S. Makino, T. Lee, and H. Sawada. *Blind Speech Separation*. Springer, 2010.

[167] M. Matassoni, R. Astudillo, A. Katsamanis, and M. Ravanelli. The DIRHA-GRID corpus: baseline and tools for multi-room distant speech recognition using distributed microphones. In *Proc. of Interspeech*, pages 1616–1617, 2014.

[168] M. Matassoni, M. Omologo, D. Giuliani, and P. Svaizer. Hidden Markov model training with contaminated speech material for

distant-talking speech recognition. *Computer Speech & Language*, 16(2):205–223, 2002.

[169] K. Matsuoka. Noise injection into inputs in back-propagation learning. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(3):436–440, 1992.

[170] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133, 1943.

[171] L. McNally, S. P. Brown, and A. L. Jackson. Cooperation and the evolution of intelligence. *Proceedings of the Royal Society of London B: Biological Sciences*, 2012.

[172] T. Menne, J. Heymann, A. Alexandridis, K. Irie, A. Zeyer, M. Kitza, P. Golik, I. Kulikov, L. Drude, R. Schlüter, H. Ney, R. Haeb-Umbach, and A. Mouchtaris. The RWTH/UPB/FORTH System Combination for the 4th CHiME Challenge Evaluation. In *CHiME 4 challenge*, pages 39–44, 2016.

[173] A. Mesaros, T. Heittola, A. Eronen, and T. Virtanen. Acoustic event detection in real life recordings. In *Proc. of EUSIPCO*, 2010.

[174] B.T. Meyer and B. Kollmeier. Optimization and Evaluation of Gabor feature sets for ASR. In *Proc. of Interspeech*, pages 906–909, 2008.

[175] Y. Miao, M. Gowayyed, and F. Metze. EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding. In *Proc. of ASRU*, pages 167–174, 2015.

[176] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur. Recurrent neural network based language model. In *Proc. of Interspeech*, pages 1045–1048, 2010.

[177] M. Mimura, S. Sakai, and T. Kawahara. Joint optimization of denoising autoencoder and dnn acoustic model based on multi-target learning for noisy speech recognition. In *Proc. of Interspeech*, 2016.

[178] S. Mirsamadi and J. Hansen. A study on deep neural network acoustic model adaptation for robust far-field speech recognition. In *Proc. of Interspeech*, pages 2430–2434, 2015.

[179] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. Playing atari with deep reinforcement learning. In *Proc. of NIPS*. 2013.

[180] M. Mohri. Finite-state transducers in language and speech processing. *Computational Linguistics*, 23(2):269–311, 1997.

[181] B. Monson, E. Hunter, and B. Story. Horizontal directivity of low- and high-frequency energy in speech and singing. *Journal of Acoustical Society of America*, 132:433, 2012.

[182] T. Moon, H. Choi, H. Lee, and I. Song. RNNDROP: A novel dropout for RNNS in ASR. In *Proc. of ASRU*, pages 65–70, 2015.

[183] D. Mostefa, N. Moreau, K. Choukri, G. Potamianos, S. Chu, A. Tyagi, J. Casas, J. Turmo, L. Cristoforetti, F. Tobia, A. Pnevmatikakis, V. Mylonakis, F. Talantzis, S. Burger, R. Stiefelhagen, K. Bernardin, and C. Rochet. The CHIL Audiovisual Corpus for Lecture and Meeting Analysis inside Smart Rooms. *Language resources and evaluation*, 41(3):389–407, 01/2008 2007.

[184] J. M. Naik, L. P. Netsch, and G. R. Doddington. Speaker verification over long distance telephone lines. In *Proc. of ICASSP*, pages 524–527, 1989.

[185] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proc. of ICML*, pages 807–814, 2010.

[186] A. Narayanan and D. Wang. Joint noise adaptive training for robust automatic speech recognition. In *Proc. of ICASSP*, pages 4380–4384, 2014.

[187] P. A. Naylor and N. D. Gaubitch. *Speech Dereverberation*. Springer, 2010.

[188] F. Nesta, T. Wada, and B. Juang. Batch-online semi-blind source separation applied to multi-channel acoustic echo cancellation. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(3):583–599, 2011.

[189] N. J. Nilsson. *The Quest for Artificial Intelligence*. Cambridge University Press, 2009.

[190] H. F. Olson and H. Belar. Phonetic typewriter. *Journal of Acoustic Society of America*, 28(6):1072–1081, 1956.

[191] M. Omologo. A prototype of distant-talking interface for control of interactive TV. In *Proc. of ASILOMAR*, pages 1711–1715, 2010.

[192] M. Omologo and P. Svaizer. Acoustic event localization using a crosspower-spectrum phase based technique. In *Proc. of ICASSP*, pages 273–276, 1994.

[193] M. Omologo and C. Zieger. Comparison between Subband and fullband NLMS for in-Car audio compensation and handsfree speech recognition. In *Proc. of IWAENC 2005*, pages 189–192, 2005.

[194] S. Hiller P. Bagshaw and M. Jack. Enhanced pitch tracking and the processing of F0 contours for computer aided intonation teaching. In *Proc, of Eurospeech*, 1993.

[195] D. Palaz, M. Magimai-Doss, and R. Collobert. Convolutional neural networks-based continuous speech recognition using raw speech signal. In *Proc. of ICASSP*, pages 4295–4299, 2015.

[196] J. Pan, C. Liu, Z. Wang, Y. Hu, and H. Jiang. Investigation of deep neural networks (dnn) for large vocabulary continuous speech recognition: Why dnn surpasses gmms in acoustic modeling. In *Proc. of ISCSLP*, pages 301–305, 2012.

[197] R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. In *Proc. of ICML*, pages 1310–1318, 2013.

[198] D. B. Paul and J. M. Baker. The design for the wall street journal-based csr corpus. In *Proc. of the Workshop on Speech and Natural Language*, 1992.

[199] K. Pearson. Mathematical Contributions to the Theory of Evolution. III. Regression, Heredity, and Panmixia. *Proceedings of the Royal Society of London, Philosophical Transactions of the Royal Society*, 187:253–318, January 1886.

[200] V. Peddinti, D. Povey, and S. Khudanpur. A time delay neural network architecture for efficient modeling of long temporal contexts. In *Proc of Interspeech*, pages 3214–3218, 2015.

[201] R. Pieraccini. *The Voice in the Machine: Building Computers that Understand Speech*. MIT press, 2012.

[202] G. Pironkov, S. Dupont, and T. Dutoit. Multi-Task Learning for Speech Recognition: An Overview. In *Proc. of ESANN*, 2016.

[203] C. Plahl, R. Schlüter, and H. Ney. Hierarchical Bottle-Neck Features for LVCSR. In *Proc. of Interspeech*, 2010.

[204] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely. The Kaldi Speech Recognition Toolkit. In *Proc. of ASRU*, 2011.

[205] O. Press, A. Bar, B. Bogin, J. Berant, and L. Wolf. Language generation with recurrent generative adversarial networks without pre-training. *CoRR*, abs/1706.01399, 2017.

[206] S. Pruzansky. Pattern-matching procedure for automatic talker recognition. *Journal of the Acoustical Society of America*, 35:354–358, 1963.

[207] S. Pruzansky and M. V. Mathews. Talker recognition procedure based on analysis of variance. *Journal of the Acoustical Society of America*, 36:2041–2047, 1964.

[208] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, February 1989.

[209] L. Rabiner and B. Huang. *Fundamentals of Speech Recognition*. Prentice-Hall, 1993.

[210] A. Ragni, K. Knill, S. Rath, and M. Gales. Data augmentation for low resource languages. In *Proc. of Interspeech*, pages 5582–5586, 2014.

[211] S. P. Rath, D. Povey, K. Vesel, and J. Cernock. Improved feature processing for deep neural networks. In *Proc. of Interspeech*, pages 109–113, 2013.

[212] M. Ravanelli, P. Brakel, M. Omologo, and Y. Bengio. Batch-normalized joint training for dnn-based distant speech recognition. In *Proc. of SLT*, pages 28–34, 2016.

[213] M. Ravanelli, P. Brakel, M. Omologo, and Y. Bengio. Improving speech recognition by revising gated recurrent units. In *Proc. of Interspeech*, 2017.

[214] M. Ravanelli, P. Brakel, M. Omologo, and Y. Bengio. Light Gated Recurrent Units for Speech Recognition. *IEEE Transactions on Emerging Topics in Computational Intelligence (under publication)*, 2017.

[215] M. Ravanelli, P. Brakel, M. Omologo, and Y. Bengio. A network of deep neural networks for distant speech recognition. In *Proc. of ICASSP*, pages 4880–4884, 2017.

[216] M. Ravanelli, L. Cristoforetti, R. Gretter, M. Pellin, A. Sosi, and M. Omologo. The DIRHA-English corpus and related tasks for distant-speech recognition in domestic environments. In *Proc. of ASRU*, pages 275–282, 2015.

[217] M. Ravanelli, V.H. Do, and A. Janin. TANDEM-Bottleneck Feature Combination using Hierarchical Deep Neural Networks. In *Proc. of ISCSLP*, pages 113–117, 2014.

[218] M. Ravanelli, B. Elizalde, K. Ni, and G. Friedland. Audio concept classification with hierarchical deep neural networks. In *Proc. of EUSIPCO*, 2014.

[219] M. Ravanelli and M. Omologo. On the selection of the impulse responses for distant-speech recognition based on contaminated speech training. In *Proc. of Interspeech 2014*, pages 1028–1032.

[220] M. Ravanelli and M. Omologo. Deliverable D2.1_D2.3_D2.4: DIRHA-simcorpora I and II. Technical report, DIRHA Consortium, 2014.

[221] M. Ravanelli and M. Omologo. Contaminated speech training methods for robust DNN-HMM distant speech recognition. In *Proc. of Interspeech*, pages 756–760, 2015.

[222] M. Ravanelli and M. Omologo. An asymmetric context window for dnn-based distant speech recognition. *Computer Speech and Language (Under review)*, 2017.

[223] M. Ravanelli and M. Omologo. Realistic impulse responses for distant speech recognition. *Speech Communication (Under review)*, 2017.

[224] M. Ravanelli, A. Sosi, M. Matassoni, M. Omologo, M. Benetti, and G. Pedrotti. Distant-talking speech recognition in surgery room : the DOMHOS project. In *Proc. of AISV*, 2013.

[225] M. Ravanelli, A. Sosi, P. Svaizer, and M. Omologo. Impulse response estimation for robust speech recognition in a reverberant environment. In *Proc. of EUSIPCO 2012*, pages 1668–1672.

[226] M. Ravanelli, P. Svaizer, and M. Omologo. Realistic multi-microphone data simulation for distant speech recognition. In *Proc. of Interspeech*, pages 2786–2790, 2016.

[227] S. Renals, T. Hain, and H. Bourlard. Interpretation of Multiparty Meetings the AMI and Amida Projects. In *Proc. of HSCMA*, pages 115–118, 2008.

[228] D. A. Reynolds and R. C. Rose. Robust text-independent speaker identification using Gaussian mixture speaker models. *IEEE Transactions on Speech and Audio Processing*, 3(1):72–83, January 1995.

[229] F. Richardson, D. A. Reynolds, and N. Dehak. Deep neural network approaches to speaker and language recognition. *IEEE Signal Process. Letters*, 22(10):1671–1675, 2015.

[230] M. Robins. The future of deep learning: Challenges & solutions. In *Proc. of the CF*, 2017.

[231] M. A. Roch, R. R. Hurtig, T. Huang, J. Liu, and S. M. Arteaga. Foreground auditory scene analysis for hearing aids. *Pattern Recognition Letters*, 28(11):1351 – 1359, 2007.

[232] I. Rodomagoulakis, P. Giannoulis, Z. I. Skordilis, P. Maragos, and G. Potamianos. Experiments on far-field multichannel speech processing in smart homes. In *Proc. of DSP*, pages 1–6, 2013.

[233] A. Romero, N. Ballas, Samira Ebrahimi K., A. Chassang, C. Gatta, and Y. Bengio. Fitnets: Hints for thin deep nets. In *Proc. of ICLR*, 2015.

[234] A. E. Rosenberg and F. K. Soong. Evaluation of a vector quantization talker recognition system in text independent and text dependent models. *Computer Speech and Language*, 22:143–157, 1987.

[235] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, pages 65–386, 1958.

[236] N. Ruiz, A. Bisazza, F. Brugnara, D. Falavigna, D. Giuliani, S. Jaber, R. Gretter, and M. Federico. FBK-IWSLT 2011. In *Proc. of IWSLT*, 2011.

[237] D. Rumelhart, G. Hinton, and R. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[238] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Neurocomputing: Foundations of research. chapter Learning Representations by Back-propagating Errors, pages 696–699. 1988.

[239] N. Ryant, M. Liberman, and J. Yuan. Speech Activity Detection on YouTube Using Deep Neural Networks. In *Proc. of Interspeech*, pages 728–731, 2013.

[240] T. N. Sainath and C. Parada. Convolutional neural networks for small-footprint keyword spotting. In *Proc. of Interspeech*, pages 4087–4091, 2015.

[241] T. N. Sainath, R. J. Weiss, A. W. Senior, K. W. Wilson, and O. Vinyals. Learning the speech front-end with raw waveform CLDNNs. In *Proc. of Interspeech*, 2015.

[242] T. N. Sainath, R. J. Weiss, K. W. Wilson, A. Narayanan, M. Bacchiani, and A. Senior. Speaker localization and microphone spacing invariant acoustic modeling from raw multichannel waveforms. In *Proc. of ASRU*, 2015.

[243] T.N. Sainath, O. Vinyals, A. Senior, and H. Sak. Convolutional, long short-term memory, fully connected deep neural networks. In *Proc. of ICASSP*, pages 4580–4584, 2015.

[244] H. Sak, A. W. Senior, and F. Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *Proc. of Interspeech*, pages 338–342, 2014.

[245] H. Sakoe and S. Chiba. Dynamic programming algorithm quantization for spoken word recognition. *IEEE Transaction on Acoustics, Speech and Signal Processing*, 4(1):43–49, 1978.

[246] R. Schluter, I. Bezrukov, H. Wagner, and H. Ney. Gammatone features and feature combination for large vocabulary speech recognition. In *Proc. of ICASSP*, volume 4, pages 649–652, 2007.

[247] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015.

[248] M. Schmidt and H. Gish. Speaker identification via support vector classifiers. In *Proc. of ICASSP*, pages 105–108, 1996.

[249] M.R Schroeder. Diffuse sound reflection by maximum-length sequences. In *Journal of Acoustic Society of America*, volume 57(1), pages 149–150, 1975.

[250] A. Schwarz, C. Huemmer, R. Maas, and W. Kellermann. Spatial Diffuseness Features for DNN-Based Speech Recognition in Noisy and Reverberant Environments. In *Proc. of ICASSP*, pages 4380–4384, 2015.

[251] A. Sehr, C. Hofmann, R. Maas, and W. Kellermann. Multi-style training of HMMS with stereo data for reverberation-robust speech recognition. In *Proc. of HSCMA 2011*, pages 196–200.

[252] F. Seide and A. Agarwal. Cntk: Microsoft's open-source deep-learning toolkit. In *Proceedings of ACM SIGKDD*, pages 2135–2135, 2016.

[253] M. Seltzer and J. Droppo. Multi-task learning in deep neural networks for improved phoneme recognition. In *Proc. of ICASSP*, pages 6965–6969, 2014.

[254] A. W. Senior, G. Heigold, M. Bacchiani, and H. Liao. GMM-free DNN acoustic model training. In *Proc. of ICASSP*, pages 5602–5606, 2014.

[255] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, NY, USA, 2004.

[256] N. S. Sohn, J. Kim and W. Sung. A statistical model-based voice activity detection. *IEEE Signal Processing Letters*, 6(1):1–3, Jan 1999.

[257] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[258] R. K. Srivastava, K. Greff, and J. Schmidhuber. Highway Networks. *CoRR*, abs/1505.00387, 2015.

[259] G. Stemmer and F. Brugnara. Integration of Heteroscedastic Linear Discriminant Analysis (HLDA) Into Adaptive Training. In *Proc. of ICASSP*, 2006.

[260] A. Stolcke, L. Ferrer, S. Kajarekar, E. Shriberg, and A. Venkataraman. MLLR transforms as features in speaker recognition. In *Proc. Eurospeech*, pages 2425–2428, 2005.

[261] I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the importance of initialization and momentum in deep learning. In *Proc. of ICML*, pages 1139–1147, 2013.

[262] R. S. Sutton and A. G. Barto. *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1998.

[263] P. Swietojanski, A. Ghoshal, and S. Renals. Hybrid acoustic models for distant and multichannel large vocabulary speech recognition. In *Proc. of ASRU*, pages 285–290, 2013.

[264] A. Temko, R. Malkin, C. Zieger, D. Macho, C. Nadeu, and M. Omologo. Clear evaluation of acoustic event detection and classification systems. *Multimodal Technologies for Perception of Humans*, pages 311–322, 2007.

[265] A. Temko and C. Nadeu. Classification of acoustic events using SVM-based clustering schemes. *Pattern Recognition*, 39(4):682–694, 2006.

[266] A. Temko, C. Nadeu, D. Macho, R. Malkin, C. Zieger, and M. Omologo. Acoustic event detection and classification. In *Computers in the Human Interaction Loop*, pages 61–73. Springer London, 2009.

[267] Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May 2016.

[268] S. Thomas, S. Ganapathy, G. Saon, and H. Soltau. Analyzing convolutional neural networks for speech activity detection in mismatched acoustic conditions. In *Proc. of ICASSP*, pages 2519–2523, 2014.

[269] L. Toth. Modeling long temporal contexts in convolutional neural network-based phone recognition. In *Proc. of ICASSP*, pages 4575–4579, 2015.

[270] M. Tu and X. Zhang. Speech enhancement based on deep neural networks with skip connections. In *Proc. of ICASSP*, pages 5565–5569, 2017.

[271] R. Tucker. Voice activity detection using a periodicity measure. *IEEE transaction on Speech and Vision*, 139(4):377–380, August 1992.

[272] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. In *Arxiv*, 2016.

[273] K. Veselý, A. Ghoshal, L. Burget, and D. Povey. Sequence-discriminative training of deep neural networks. In *Proc. of Interspeech*, pages 2345–2349, 2013.

[274] K. Vesely, M. Karafiat, and F. Grezl. Convolutive Bottleneck Network features for LVCSR. In *Proc. of ASRU*, pages 42–47, 2011.

[275] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11:3371–3408, 2010.

[276] T. K. Vintsyuk. Speech discrimination by dynamic programming. *Kibernetika*, 4(2):81–88, 1968.

[277] A. J. Viterbi. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Transactions on Information Theory*, 13:260–269, 1967.

[278] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. In *Proc. of NIPS*, pages 613–621, 2016.

[279] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang. Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech, and Signal processing*, 37(3):328–339, 1989.

[280] L. Wan, M. D. Zeiler, S. Zhang, Y. LeCun, and R. Fergus. Regularization of neural networks using dropconnect. In *Proc. of ICML*, pages 1058–1066, 2013.

[281] S. Wang and C. Manning. Fast dropout training. In *Proc. of ICML*, pages 118–126, 2013.

[282] Y. Wang, J. Li, and Y. Gong. Small-footprint high-performance deep neural network-based speech recognition using split-VQ. In *Proc. of ICASSP*, pages 4984–4988, 2015.

[283] Z. Q. Wang and D. Wang. A joint training framework for robust automatic speech recognition. *IEEE/ACM Trans. Audio, Speech & Language Processing*, 24(4):796–806, 2016.

[284] F. Weninger, H. Erdogan, S. Watanabe, E. Vincent, J. Le Roux, J. R. Hershey, and B. W. Schuller. Speech enhancement with LSTM recurrent neural networks and its application to noise-robust ASR. In *Proc. of LVA/ICA*, pages 91–99, 2015.

[285] F. Weninger, S. Watanabe, J. Le Roux, J.R. Hershey, Y. Tachioka, J. Geiger, B. Schuller, and G. Rigoll. The MERL/MELCO/TUM System for the REVERB Challenge Using Deep Recurrent Neural Network Feature Enhancement. In *Proc. of IEEE REVERB Workshop*, 2014.

[286] B. Widrow and M. E. Hoff. Adaptive switching circuits. In *Proc. of IRE WESCON Convention Record*, pages 96–104, 1960.

[287] M. Wolf and C. Nadeu. On the potential of channel selection for recognition of reverberated speech with multiple microphones. In *Proc. of Interspeech*, pages 574–577, 2010.

[288] M. Wolf and C. Nadeu. Channel selection measures for multi-microphone speech recognition. *Speech Communication*, 57:170–180, February 2014.

[289] C. Wolfel, M Fugen, S. Ikbal, and J. McDonough. Multi-source far-distance microphone selection and combination for automatic transcription of lectures. In *Proc. of Interspeech*, pages 1121–1125, 2006.

[290] M. Wölfel and J. McDonough. *Distant Speech Recognition*. Wiley, 2009.

[291] D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.

[292] X. Xiao, S. Zhao, D. Nguyen, X. Zhong, D. Jones, E. Chng, and H. Li. Speech dereverberation for enhancement and recognition using dynamic features constrained deep neural networks and feature adaptation. *EURASIP J. Adv. Sig. Proc.*, 2016:4, 2016.

[293] Y. Xu, J. Du, L. Dai, and C. Lee. An experimental study on speech enhancement based on deep neural networks. *IEEE Signal Processing Letters*, 21(1):65–68, 2014.

[294] Y. Xu, J. Du, L. R. Dai, and C. H. Lee. A regression approach to speech enhancement based on deep neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(1):7–19, Jan 2015.

[295] T. Yoshioka, N. Ito, M. Delcroix, A. Ogawa, K. Kinoshita, M. Fujimoto, C. Yu, W. J. Fabian, M. Espi, T. Higuchi, S. Araki, and T. Nakatani. The NTT CHiME-3 system: Advances in speech enhancement and recognition for mobile multi-microphone devices. In *Proc. ASRU*, pages 436–443, 2015.

[296] S. Young et al. *HTK – Hidden Markov Model Toolkit*, 2006.

[297] D. Yu and L. Deng. *Automatic Speech Recognition - A Deep Learning Approach*. Springer, 2015.

[298] D. Yu, K. Yao, H. Su, G. Li, and F. Seide. KL-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition. In *Proc. of ICASSP*, pages 7893–7897, 2013.

[299] M. Zeiler. ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701, 2012.

[300] M. D. Zeiler, M. Ranzato, R. Monga, M. Mao, K. Yang, Q. V. Le, P. Nguyen, A. Senior, V. Vanhoucke, J. Dean, and G. E. Hinton. On rectified linear units for speech processing. In *Proc. of ICASSP*, pages 3517–3521, 2013.

[301] X. L. Zhang and J. Wu. Deep belief networks based voice activity detection. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(4):697–710, 2013.

[302] Y. Zhang, G. Chen, D. Yu, K. Yao, S. Khudanpur, and J. R. Glass. Highway long short-term memory RNNS for distant speech recognition. In *Proc. of ICASSP*, pages 5755–5759, 2016.

[303] G. Zhou, J. Wu, C. Zhang, and Z. Zhou. Minimal gated unit for recurrent neural networks. *CoRR*, abs/1603.09420, 2016.

[304] Q. Zhu, A. Stolcke, B. Y. Chen, and N. Morgan. Using MLP features in SRIs conversational speech recognition system. In *Proc. of Interspeech*, pages 2141–2144, 2005.

[305] C. Zieger and M. Omologo. Acoustic Event Classification Using a Distributed Microphone Network with a GMM/SVM Combined Algorithm. In *Proc. of Interspeech*, pages 504–5164, 2008.

[306] E. Zwyssig, M. Ravanelli, P. Svaizer, and M. Omologo. A multichannel corpus for distant-speech interaction in presence of known interferences. In *Proc. of ICASSP*, pages 4480–4484, 2015.

Appendices

Appendix A

Multi-Microphone Setups

Many experiments conducted in this thesis are performed in a domestic scenario. The reference environment is a real apartment (referred to as DIRHA apartment) that was available for experiments under the DIRHA project[1]. This apartment have been equipped with various microphone configurations, that will be described in the following sections:

A.1 Microphone Configuration 1 (MC-1)

The first setup was based on 40 microphones distributed in five rooms of the DIRHA apartment (i.e., living-room, kitchen, corridor,bathroom, and bedroom). The sensors were high-quality omnidirectional microphones (Shure MX391/O), connected to multichannel clocked pre-amp and A/D boards (RME Octamic II), which allowed a perfectly synchronous sampling at 48 kHz, with 24 bit resolution. Bathroom, corridor and bedroom were equipped with a limited number of microphone pairs and triplets (i.e., overall 12 microphones), while the living-room and the kitchen comprise the largest concentration of sensors and devices. As shown in Figure A.1, the living-room includes three microphone pairs, a microphone triplet, and 6-microphone ceiling arrays. This setup was adopted for real recordings

[1] https://dirha.fbk.eu/

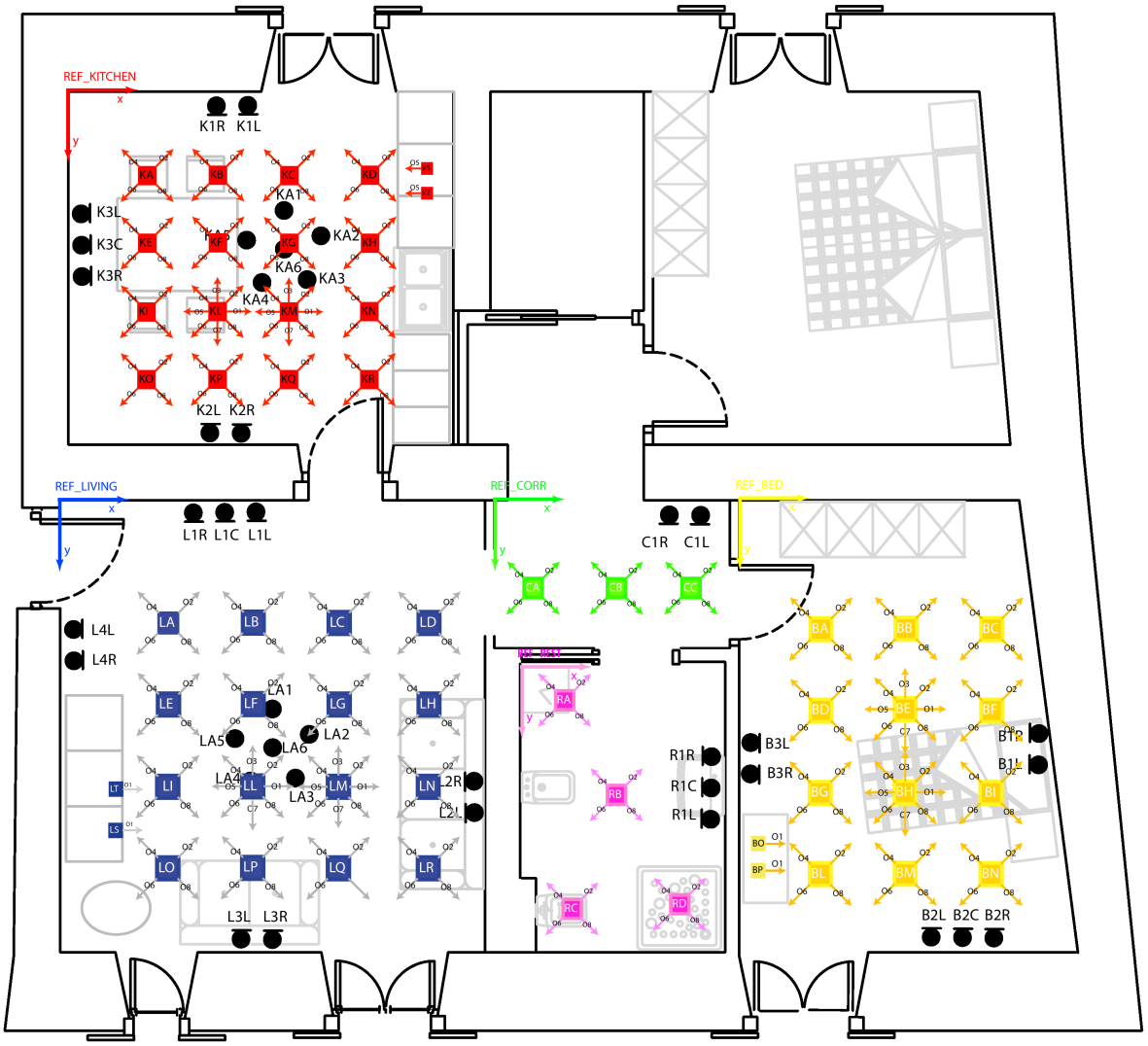


Figure A.1: Floor-plan of the DIRHA apartment equipped with a multi-room microphone network.

as well as for the extensive multi-room multi-microphone IR measurement champaign (see DIRHA-IRs Collection) used for the development of the DIRHA-Sim Corpora [51] described in the following.

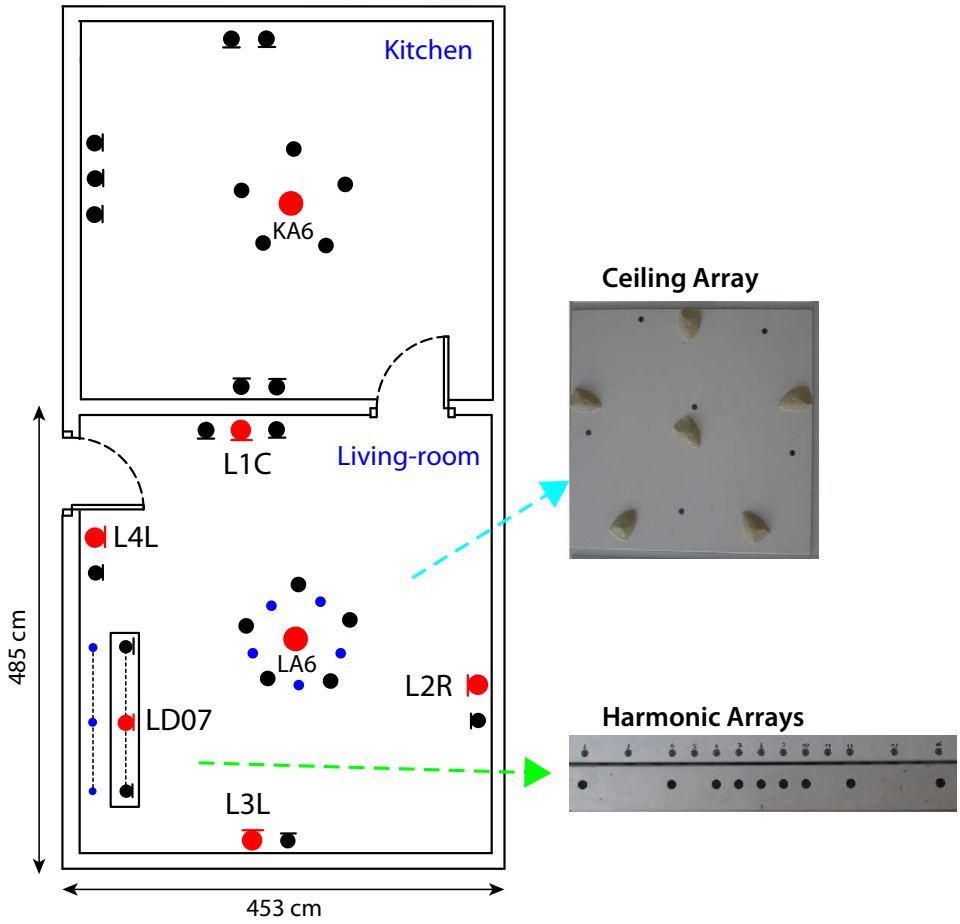


Figure A.2: An outline of the microphone set-up adopted for the DIRHA-ENGLISH corpus. Blue small dots represent digital MEMS microphones, red ones refers to the channels used for most of the experiments, while black ones represent the other available microphones. The right pictures show the ceiling array and the two linear harmonic arrays installed in the living-room.

A.2 Microphone Configuration 2 (MC-2)

A second multi-microphone setup have been considered to explore a different microphone configuration (see Figure A.2), which comprises 60 microphones distributed in the living-room (47) and the kitchen (13) of the DIRHA apartment. In particular, the living-room was equipped with:

- a linear harmonic array composed of 13 Electrect microphones

- a linear harmonic array composed of 13 MEMS microphones
- a circular array composed of 6 Shure MX391/0 microphones
- a circular array composed of 6 MEMS microphones
- 9 Shure MX391/0 (3 pairs and 1 triplet) distributed in the four walls of the livingroom

Shure and Electrect microphones were connected to multichannel clocked pre-amp and A/D boards (RME Octamic II), which allowed a perfectly synchronous sampling at 48 kHz, with 24 bit resolution. The MEMS microphones were developed by ST-Microelectronics and the recordings have been conducted with their original acquisition board.

This setup was adopted for the real recordings of the DIRHA-English Dataset. Moreover an IR measurement champaign (see DIRHA-IRs Collection) was conducted, and the measured IRs were used for generating the simulated part of the DIRHA-English Dataset. With this microphone configuration several hours of various noise sequences typical of a domestic environment were also recorded. The noise sequences were used to both add realistic noise in the simulations of the DIRHA-English Dataset and to contaminate training datasets (such as the WSJ corpus) with also additive noise.

This setup, that is limited to the kitchen and the livingroom, is depicted in Figure A.2. The living-room includes three microphone pairs, a microphone triplet, and 6-microphone ceiling arrays

It also considers harmonic arrays and MEMS microphones which were unavailable in the previous setup. This setup was adopted for the development of the DIRHA-English Corpus [216].

Appendix B

Corpora

The experimental evaluations of the techniques proposed in this thesis have performed on different datasets, that are listed and briefly described in the following.

- **APASCI**: It is an Italian high-quality speech database recorded in a recording studio under close-talking conditions [5]. It includes 5,290 phonetically rich sentences and about 350 minutes of speech. The speech material was read by 100 Italian speakers (50 male and 50 female). Each of them uttered 1 calibration sentence, 4 sentences with a wide phonetic coverage, 15 or 20 sentences with a wide diphonic coverage. Reverberated version of APASCI (denoted as *APASCI-rev*) have been generated with the approach described in Chapter 4 to train acoustic models of a speech recognizer. The versions denoted as *APASCI-rev&noise* also includes the contribution of additive noise that was recorded in the DIRHA apartment.

- **CHiME 4**: This dataset [172] is based on both real and simulated data recorded in four noisy environments (on a bus, cafe, pedestrian area, and street junction). The training set is composed of 43690 noisy WSJ sentences (75 hours) recorded by five microphones (arranged on a tablet) and uttered by a total of 87 speakers. The development set

(DT) is based on 3280 WSJ sentences uttered by four speakers (1640 are real utterances referred to as DT-real, and 1640 are simulated denoted as DT-sim). The test set (ET) is based on 1320 real utterances (ET-real) and 1320 simulated sentences (DT-real) from other four speakers. The experiments reported in this thesis are based on the single channel setting, in which the test phase is carried out with a single microphone (randomly selected from the considered microphone setup). More information on CHiME data can be found in [10].

- **DIRHA-IRs Collection**: During the DIRHA project several efforts have been devoted to measure realistic impulse responses. A first IR measurement campaign has been carried out with the network of 40 microphones described in App. A.1. The IRs measurement process explored 57 different positions with at least four orientations for each position. In total, more than 9000 IRs have been measured. Cross-room impulse responses are also included in order to simulate cross-room audio propagation effects. The IRs measurements were based on a professional studio monitor (Genelec 8030A) with the methodology described in Chapter 4. The Time of Flight (TOF) information, crucial to applications such as acoustic event localization, beam-forming, and multi-microphone signal processing in general, has been preserved by means of six sample-synchronized multi-channel audio cards. The IRs were measured at 48kHz sampling frequency with 16-bit accuracy. A second IR measurement campaign was conducted in the same apartment with the same methodology, but with the microphone configuration described in App. A.2.

- **DIRHA-English**: This corpus is multi-microphone dataset [216], recently realized under the EC DIRHA project. The reference scenario is the DIRHA apartment equipped with the microphone configuration

described in App. A.2. The overall dataset is composed of 1-minute sequences, comprising phonetically-rich, conversational speech, keywords, commands, and wsj sentences. The corpus includes both real and simulated data. Simulations have been generated with the approach discussed in Chapter 4 by combining high-quality close-talking recordings with impulse responses measured in the targeted environment. Background noise recorded in the apartment was also added to the simulations. Together with simulated data, real recording in different positions of the living-room are performed. The corpus includes 12 US and 12 UK English native speakers. Different subsets have been derived from the overall corpus and used in this thesis. For instance, the part denoted in the manuscript as *DIRHA-English-WSJ5k* contains a collection of the wsj-5k sentences uttered by the aforementioned US speakers. Six speakers are part of the set1 subpart (409 sentences), while the other speakers belong to the set2 (330 sentence). The sentences are available in various scenarios of increasing complexity, i.e. close-talking *DIRHA-English-WSJ5k (close-talk)*, reverberated *DIRHA-English-WSJ5k (rev)*, reverb+noise conditions *DIRHA-English-WSJ5k (rev&noise)*. The WSJ part of the dataset in being distributed by the Linguistic Data Consortium (LDC), while the phonetically-rich sentences can be downloaded from the DIRHA website. The DNN baselines obtained with Kaldi (using the standard Karel's recipe) in the aforementioned conditions are reported in Table B.1 for reference purposes (fMLLR features). The ASR performance is reported in terms of Word Error Rate (%) and the reference microphone is $LA6$[1].

- **DIRHA-SimCorpora**: The database was collected with the mi-

[1]See our github page `https://github.com/SHINE-FBK/DIRHA_English_wsj` for a complete overview on the ASR baselines.

Dataset. / *Condition*	set1		set2	
	Sim	Real	Sim	Real
Close-Talking	-	3.7	-	2.2
Distant-Talking (Rev)	16.2	13.6	9.8	9.4
Distant-Talking (Rev&Noise)	22.8	30.0	25.0	16.6

Table B.1: DNN baselines for the DIRHA-English-WSJ obtained with kaldi (karel's recipe).

crophone setup described in App. A.1. The DIRHA SimCorpus is a multi-microphone, multi-language (Italian, German, Portuguese, Greek) and multi-room database containing simulated acoustic sequences derived from the DIRHA apartment. For each language, the corpus contains a set of acoustic sequences of duration 60 seconds, at 48kHz sampling frequency and 16-bit accuracy. Each sequence consists of real background noise with superimposed various localized acoustic events. Acoustic events occur randomly (and rather uniformly) in time and in space (within predefined positions) with various dynamics. The acoustic wave propagation from the sound source to each single microphone is simulated by convoluting the clean signals with the respective impulse response (IR). Acoustic events are divided into two main categories, i.e., speech and non-speech. Speech events include different types of sentences (i.e., phonetically-rich sentences, read, commands, spontaneous speech and commands, keywords) uttered by different speakers in the four languages. Non-speech events have been selected from a collection of high-quality sounds typically occurring within a home environment (e.g., radio, TV, appliances, knocking, ringing, creaking, etc.). For cross-language comparison purposes, each simulated acoustic sequence has been replicated in the four languages while preserving the same background noises and non-speech sources. Gender and timing of the active speakers have been

preserved across the different languages, in order to further ensure homogeneity. Data were generated by means of a multi-microphone simulation framework developed at FBK. A subset of this corpus has been delivered in the contest of HSCMA 2014 [37] and EVALITA 2014 [36].

- **DIRHA-GRID**: This corpus [167] is a multi-microphone and multi-room simulated database containing a set of acoustic scenes of 1-minute duration (at 16kHz sampling frequency and 16-bit accuracy) which are observed by the microphone network described in App. A.1. Each acoustic scene is composed of both speech, i.e., short English commands from the GRID database [60], and non-speech acoustic sources, i.e., typical home noises. Each acoustic event (both speech and non-speech) occurs randomly in time and in space and can take place in any of the microphone-equipped rooms. In particular, a variable number of short commands (ranging from 4 to 7) arise in each 1-minute long acoustic scene. An overlap in time between speech and non-speech sources is possible, while an overlap between speech sources cannot occur. The corpus is divided into 3 chunks (dev 1, test1, test2) containing 75 acoustic sequences each with 12 different speakers (6 male and 6 female) involved for each dataset. Data were generated by means of a multi-microphone simulation framework developed at FBK and can be downloaded from the DIRHA website.

- **DIRHA-AEC**: The database was collected in the DIRHA apartment using the microphone setup described in App. A.1. In contrast to the other real databases recorded under the DIRHA project, the DIRHA AEC corpus is specifically designed for Acoustic Echo Cancellation (AEC) and is currently available in Italian. In particular, it is based on real recordings involving 13 speakers (6 males and 7 females) that

read a list of 50 commands in five different positions/orientations in the living- room. A set of simulated sequences were also generated, in which all the commands were repeated with no overlapping sources (Real-S0), an overlap with a prompt (Real-S1), an overlap with the television (Real-S2) and with both overlaps (Real- S3). In total, 2600 speech commands were recorded.

- **Euronews**: Data come from the portal Euronews and were acquired both from the Web and from TV [101]. The overall corpus includes data in 10 languages (Arabic, English, French, German, Italian, Polish, Portuguese, Russian, Spanish and Turkish) and was designed both to train AMs and to evaluate ASR performance. For each language, the corpus is composed of about 100 hours of speech for training and about 4 hours, manually transcribed, for testing. Training data include the audio, some reference text, the ASR output and their alignment. The portion used in this thesis is the Italian part of the dataset. Reverberated version of Euronews (denoted as *Euronews-rev*) have been generated with the approach described in Chapter 4 to train acoustic models of a speech recognizer.

- **TED-talks corpus**: This dataset was released in the context of the IWSLT evaluation campaigns [72]. The training set is composed of 820 talks with a total of about 166 hours of speech. The development test is composed of 81 talks (16 hours), while the test sets (TST 2011 and TST 2012) are based on 8 talks (1.5 hours) and 32 talks (6.5 hours), respectively.

- **TIMIT**: This corpus contains 16kHz speech recording of 630 American English speakers, each reading 10 phonetically-rich sentences. TIMIT is based on about 5 hours of speech recordings. Reverberated version of TIMIT (denoted as *TIMIT-rev*) have been generated

with the approach described in Chapter 4 to train acoustic models of a speech recognizer. The versions denoted as *TIMIT-rev&noise* also includes the contribution of additive noise that was recorded in the DIRHA apartment.

- **Wall Street Journal (WSJ)**: The Wall Street Journal dataset (WSJ) consists of news text read by US native speakers. It is composed of two parts, often known as WSJ0 and WSJ1. The part used for this thesis is the WSJ0-5k, that is composed of 37416 sentences uttered by 283 speakers for a total of about 15 hours of speech material. Reverberated version of WSJ (denoted as *WSJ-rev*) have been generated with the approach described in Chapter 4 to train acoustic models of a speech recognizer. The versions denoted as *WSJ-rev&noise* also include the contribution of additive noise that was recorded in the DIRHA apartment.

Appendix C

Experimental Setups

We report now a detailed description of all the experimental setups adopted in the various experiments of this thesis.

C.1 Data Contamination - Setup 1

These experiments were obtained using contaminated data (for training purposes) and real data collected in the DIRHA apartment equipped with the microphone network described in App A.1. The reference language of these experiments was Italian. In particular, different recording sessions were performed in the given environment, to collect both speech material and audio signals useful to estimate impulse responses.

Speakers and microphones were both located in the living-room. The sound source was located in position LB with with frontal orientation towards the reference microphone $L3R$, that located at a distance of about 4 m (see Figure A.1).

For impulse response measurement purposes, different categories of loudspeakers were considered. In particular, the following results are based on the use of a professional studio monitor, i.e., *Genelec 8030*. In order to estimate impulse responses, MLS, LSS, and ESS excitation sequences were diffused by each loudspeaker in the environment, with varying length and

amplifying settings. Each excitation signal was preceded and followed by fade-in and fade-out sequences of 50 ms duration, in order to avoid the introduction of any possible numerical clicks. Speech data collection was then conducted with all the real speakers located in the same position where the loudspeaker was previously placed. For comparison purposes, the speech uttered by each speaker was also recorded with a professional close-talking *Countryman E6* microphone.

The speech material used to train the distant-speech recognizer consists in contaminated versions (one for each IR measurements settings) of APASCI.

In order to evaluate speech recognition performance, the utterances pronounced by 11 Italian speakers (which were not in the APASCI corpus) in the above-mentioned apartment. The speaker read a list of 125 command-and-control sentences.

The speech recognition system investigated in this work is based on a standard front-end processing consisting of a pre-emphasis step followed by feature extraction. The pre-emphasized signal is blocked and Hamming windowed into frames of 20 ms duration (with 50% overlapping). For each frame, 12 Mel-frequency Cepstral Coefficients (MFCCs) and the log-energy are extracted. MFCCs are normalized by subtracting the means, while the log-energy is normalized with respect to the maximum value on the whole utterance. The resulting normalized MFCCs and log-energy, together with their first and second order derivatives, are then arranged into a single observation vector of 39 components. Acoustic modeling is based on a GMM-HMM that operates at context-independent phone-like unit level, and is derived by applying the *Baum-Welch* algorithm, while the recognition step is accomplished by using *Viterbi* algorithm.

The Word Loop (WL) task is based on small vocabulary of 233 words used to create any of the above-mentioned command-and-control sentences.

Although the introduction of a grammar or language modeling would increase the system performance, in this work a word loop task was prefered to better emphasize any experimental evidence at acoustic level, which would otherwise be partially missed. It is worth noting that the vocabulary includes a quite large number of short and confusable words.

C.2 Data Contamination - Setup 2

The reference scenario for these experiments is the microphone configuration described in App. A.2. The considered task is the Wall Street Journal (WSJ-5k), in agreement with the task addressed in the CHiME 3 challenge. While CHiME 3 was pretty focused on robustness against noise, in this work the main source of disturbance is reverberation.

The training phase is based on the WSJ0 database (LDC catalog number LDC93S6A), which was contaminated with three impulse responses corresponding to different positions of the speaker in the targeted living-room. Note that, according to our past experience, this small number of high-quality IRs is sufficient to properly derive reverberant-robust acoustic models, as shown in our previous work [220]. The considered IRs were measured, or simulated with the image method, depending on the specific experiment.

For test purposes we employed both real and simulated data of the DIRHA English dataset (set1 portion) [216]. We actually considered only 5 of the 6 available US speakers since one of the speakers ($spk01$) was not available for the recordings in the DIRHA apartment. For the experiments reported in this work, we thus removed it from the considered dataset.

Beside real recordings, simulated data have also been created combining high-quality close-talking recordings with the measured impulse responses. With this purpose, in order to allow a fair comparison between real and

simulated data, we asked the same speakers to utter again the sentences in the FBK recording studio, which led to a very high-quality clean speech material. Moreover, for each position/orientation of the speaker in the real recordings, a corresponding IR was measured, allowing us to derive a simulated corpus well-matching in terms of speaker positions, orientations, and other signal characteristics, the real data collected in the apartment.

The experimental part of this work is based on the Kaldi toolkit [204]. The recipe considered for training and test the DSR system is similar to the s5 recipe proposed in the Kaldi release for WSJ data. In short, the speech recognizer is based on standard MFCCs and acoustic models of increasing complexity. In the following section, "*mono*" is the simplest system based on 48 context-independent phones of the English language, each modeled by a three state left-to-right HMM (overall using 1000 gaussians). A set of context-dependent models are then derived. In "*tri1*" 2.5k tied states with 15k gaussians are trained by exploiting a binary regression tree."*Tri2*"is an evolution of the standard context-dependent model in which a Linear Discriminant Analysis (LDA) is applied. In both ''*tri3*" and "*tri4*" models Speaker Adaptive Training (SAT) is also performed. The difference is that "*tri4*" is bootstrapped by the previously computed ''*tri3*" model. The considered "*DNN*", based on the Karel's recipe [204], is composed of six hidden layers of 2048 neurons, with a context window of 11 consecutive frames (5 before and 5 after the current frame) and an initial learning rate of 0.008. The weights are initialized via RBM pre-training, while the fine tuning is performed with stochastic gradient descent optimizing cross-entropy loss function. The sampling frequency for ASR experiments is 16 kHz.

C.3 Data Contamination - Setup 3

In this work, we use a Context-Dependent DNN-HMM speech recognizer, where every unit is modeled by a three state left-to-right HMM, and the tied-state observation probabilities are estimated through a DNN.

Feature extraction is based on blocking the signal into frames of 25 ms with 10 ms overlapping. For each frame, 13 MFCCs plus pitch and Probability of Voicing (PoV) are extracted. The pitch and PoV are estimated through the normalized autocorrelation method discussed in [86]. The resulting features, together with their first and second order derivatives, are then arranged into a single observation vector of 45 components. Finally, a context window gathering several consecutive frames followed by a mean and variance normalization of the feature space are applied before feeding the DNN. The DNN, trained with the Kaldi toolkit (Karel's recipe) [204], is composed of sigmoid-based hidden neurons, while the output layer is based on softmax activation functions. The pre-training phase is carried out by stacking Restricted Boltzmann Machines (RBM) [117] to form a deep belief network, while the fine-tuning is performed by a stochastic gradient descent optimizing cross-entropy loss function. In the latter phase, the initial learning rate is kept fixed as long as the increment of the frame accuracy on the dev-set is higher than 0.5%. For the following epochs, the learning rate is halved until the increment of frame accuracy is less than the stopping threshold of 0.1%. The decoding is performed by adopting a phone-loop based grammar. As previously outlined, even though the use of more complex grammars or language models is certainly helpful in increasing the recognition performance, the adoption of a simple phone-loop is due to the need of an experimental evidence not biased by a LM. In this study, as in [219], a set of 26 phone units of the Italian language was chosen for evaluation purposes

C.4 Managing Time Contexts - Setup 1

In this work the experiments were conducted in three different acoustic conditions of increasing complexity: close-talking (*Clean*), distant-talking with reverberation (*Rev*), and distant-talking with both noise and reverberation (*Rev&Noise*).

In the context of the close-talking experiments we considered the standard WSJ dataset for training, and the close-talking part of the DIRHA-English-WSJ Dataset (DIRHA-WSJ-close-talk) for test purposes (see App. B).

The reference environment for most of the experiments conducted in this study was the living-room of the DIRHA apartment, that was equipped with the microphone setup described in App. A.2.

A set of experiments was carried out to study distant-talking conditions where only reverberation acts as a source of disturbance (Rev). In these cases, training was performed using a contaminated dataset (WSJ-rev), which was generated by convolving the original WSJ data with a set of three impulse responses chosen from the aforementioned collection. Test data were based on a contaminated version of the close-talking test data (DIRHA-Englsh-WSJ5k-clean, set1). In order to simulate several speaker positions and orientations, a set of 36 impulse responses (different from those used for training) was used for the latter dataset.

To explore more challenging conditions characterized by both noise and reverberation (Rev&Noise), real recordings have also been performed. The real recordings (DIRHA-English-WSJ, real set1 portion), are part of the recently-released DIRHA English WSJ corpus [216].

To test our approach in different environments, other contaminated versions of the training and test data are generated with different impulse responses (either measured or computed with the image method [3]), as will

be discussed in Sec. 5.2.3 and Sec. 5.2.3. In this work, we use a context-dependent DNN-HMM speech recognizer, where every unit is modeled by a three state left-to-right HMM, and the tied-state observation probabilities are estimated through a DNN.

Feature extraction was based on blocking the signal into frames of 25 ms with an overlap of 10 ms. The experimental activity was conducted considering different acoustic features, i.e., 39 MFCCs (13 static+Δ+$\Delta\Delta$), 40 log-mel filter-bank features (FBANKS), as well as 40 fMLLR features (extracted as reported in the s5 recipe of Kaldi [204]). Features of consecutive frames were gathered into both symmetric and asymmetric observation windows. Mean and variance normalization of the feature space is applied before feeding the DNN.

The DNNs, trained with the Kaldi toolkit [204] (Karel's recipe), was composed of six sigmoid-based hidden layers of 2048 neurons, while the output was based on a softmax classifier. Weights were initialized with close-to-zero values with small variance. Training was performed with SGD optimizing the cross-entropy loss function, and its evolution was monitored using a small validation set (10% of the training data) that was randomly extracted from the training corpus. The performance on the validation set was monitored after each epoch to perform learning rate annealing as well as for checking the stopping condition. In particular, the initial learning rate was kept fixed at 0.008 as long as the increment of the frame accuracy on the validation was higher than 0.5%. For the following epochs, the learning rate is halved until the increment of frame accuracy was less than the stopping threshold of 0.1%. The labels for DNN training were derived from an alignment on the tied states, which was performed with a previously-trained HMM-GMM acoustic model [204] with the same input features and adopting the same corpus. For Convolutional Neural Network (CNNs) experiments, we replaced the first two fully-connected layers of the

above-mentioned DNN with two convolutional layers based on 128 and 256 filters, respectively. The decoding phase was performed with standard WSJ language model.

C.5 Managing Time Contexts - Setup 2

The architecture adopted for these experiments consisted of multiple recurrent layers, that were stacked together prior to the final softmax classifier. These recurrent layers were bidirectional RNNs [97], which were obtained by concatenating the forward hidden states (collected by processing the sequence from the beginning to the end) with backward hidden states (gathered by scanning the speech in the reverse time order). Recurrent dropout was used as regularization technique. Since extending standard dropout to recurrent connections hinders learning long-term dependencies, we followed the approach introduced in [182, 79], that tackles this issue by sharing the same dropout mask across all the time steps. Moreover, batch normalization was adopted exploiting the method suggested in [148]. The feed-forward connections of the architecture were initialized according to the *Glorot*'s scheme [89], while recurrent weights were initialized with orthogonal matrices [149]. Similarly to [212], the gain factor γ of batch normalization was initialized to 0.1 and the shift parameter β was initialized to 0.01

Before training, the sentences were sorted in ascending order according to their lengths and, starting from the shortest utterance, minibatches of 8 sentences were progressively processed by the training algorithm. This sorting approach minimizes the need of zero-paddings when forming minibatches, resulting helpful to avoid possible biases on batch normalization statistics. Moreover, the sorting approach exploits a curriculum learning strategy [18] that has been shown to slightly improve the performance

and to ensure numerical stability of gradients. The optimization was done using the Adaptive Moment Estimation (Adam) algorithm [137] running for 22 epochs (with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$). The performance on the development set was monitored after each epoch, while the learning rate was halved when the performance improvement went below a certain threshold ($th = 0.001$). Gradient truncation was not applied, allowing the system to learn arbitrarily long time dependencies.

The main hyperparameters of the model (i.e., learning rate, number of hidden layers, hidden neurons per layer, dropout factor) were optimized on the development data. In particular, we guessed some initial values according to our experience, and starting from them we performed a grid search to progressively explore better configurations. A total of 20-25 experiments were conducted for all the various RNN models. As a result, an initial learning rate of 0.0013 and a dropout factor of 0.2 were chosen for all the experiments. The optimal numbers of hidden layers and hidden neurons, instead, depend on the considered dataset/model, and range from 4 to 5 hidden layers with 375-607 neurons.

For DNN-HMM experiments, the DNN is trained to predict context-dependent phone targets. The feature extraction is based on blocking the signal into frames of 25 ms with an overlap of 10 ms. The experimental activity is conducted considering different acoustic features, i.e., 39 MFCCs (13 static+Δ+$\Delta\Delta$), 40 log-mel filter-bank features (FBANKS), as well as 40 fMLLR features (extracted as reported in the s5 recipe of Kaldi [204]). The labels were derived by performing a forced alignment procedure on the original training datasets. See the standard s5 recipe of Kaldi for more details [204]. During test, the posterior probabilities generated for each frame by the RNN are normalized by their prior probabilities. The obtained likelihoods are processed by an HMM-based decoder, that, after integrating the acoustic, lexicon and language model information in a single search

graph, finally estimates the sequence of words uttered by the speaker. The RNN part of the ASR system was implemented with Theano [267], that was coupled with the Kaldi decoder [204] to form a context-dependent RNN-HMM speech recognizer.

The models used for the CTC experiments consisted of 5 layers of bidirectional RNNs of either 250 or 465 units. Unlike in the other experiments, weight noise was used for regularization. The application of weight noise is a simplification of adaptive weight noise [95] and has been successfully used before to regularize CTC-LSTM models [98]. The weight noise was applied to all the weight matrices and sampled from a zero mean normal distribution with a standard deviation of 0.01. Batch normalization was used with the same initialization settings as in the other experiments. Glorot's scheme was used to initialize all the weights (also the recurrent ones). The input features for these experiments were 123 dimensional FBANK features (40 + energy + Δ+$\Delta\Delta$). These features were also used in the original work on CTC-LSTM models for speech recognition [98]. The CTC layer itself was trained on the 61 label set. Decoding was done using the best-path method [96], without adding any external phone-based language model.

Unlike in the other experiments, the utterances were not sorted by length.

C.6 Networks of DNNs - Setup 1

The features considered in this work are standard 39 Mel-Cepstral Coefficients (MFCCs) computed every 10 ms with a frame length of 25 ms. The speech enhancement DNN is fed with a context of 21 consecutive frames and predicts (every 10 ms) 11 consecutive frames of enhanced MFCC features. The idea of predicting multiple enhanced frames was also explored in [81]. All the layers used Rectified Linear Units (ReLU), except for the

output of the speech enhancement (linear) and the output of speech recognition (softmax). Batch normalization [123] is employed for all the hidden layers, while dropout [257] is adopted in all part of the architecture, except for the output layers.

The datasets used for joint training are obtained through a contamination of clean corpora (i.e., TIMIT and WSJ) with noise and reverberation. The labels for the speech enhancement DNN (denoted as x_{clean} in Alg.1) are the MFCC features of the original clean datasets. The labels for the speech recognition DNN (denoted as y_{lab} in Alg.1) are derived by performing a forced alignment procedure on the original training datasets. See the standard s5 recipe of Kaldi for more details [204].

The weights of the network are initialized according to the *Glorot* initialization [89], while biases are initialized to zero. Training is based on a standard Stochastic Gradient Descend (SGD) optimization with mini-batches of size 128. The performance on the development set is monitored after each epoch and the learning rate is halved when the performance improvement is below a certain threshold. The training ends when no significant improvements have been observed for more than four consecutive epochs. The main hyperparameters of the system (i.e., learning rate, number of hidden layers, hidden neurons per layer, dropout factor and λ) have been optimized on the development set.

The proposed system, which has been implemented with Theano [267], has been coupled with the Kaldi toolkit [204] to form a context-dependent DNN-HMM speech recognizer.

In order to provide an accurate evaluation of the proposed technique, the experimental validation has been conducted using different training datasets, different tasks and various environmental conditions[1].

[1]To allow reproducibility of the results reported in this section, the code of our joint-training system will be available at `https://github.com/mravanelli`. In the same repository, all the scripts needed for the data contamination will be available. The public distribution of the DIRHA-English dataset is under

The experiments with TIMIT are based on a phoneme recognition task (aligned with the Kaldi s5 recipe). The original training dataset has been contaminated with a set of realistic impulse responses measured in a real apartment. The reverberation time (T_{60}) of the considered room is about 0.7 seconds. Development and test data have been simulated with the same approach. More details about the data contamination approach can be found in [225, 51, 226].

The WSJ experiments are based on the popular wsj5k task (aligned with the CHiME 3 [10] task) and are conducted under two different acoustic conditions. For the *WSJ-Rev* case, the training set is contaminated with the same set of impulse responses adopted for TIMIT. For the *WSJ-Rev+Noise* case, we also added non-stationary noises recorded in a domestic context (the average SNR is about 10 dB). The test phase is carried out with the DIRHA English Dataset (set 2 portion), consisting of 409 WSJ sentences uttered by six native American speakers in the above mentioned apartment. For more details see [216, 226].

C.7 Networks of DNNs - Setup 2

To provide an accurate evaluation of the proposed technique, the experimental validation has been conducted using different training datasets, different tasks and various environmental conditions. In particular, a set of experiments with TIMIT has been performed to test the proposed paradigm in low-resources conditions. To validate on a more realistic task, the proposed technique has also been evaluated on a WSJ task.

The experiments with TIMIT are based on a phoneme recognition task (aligned with the Kaldi s5 recipe). The original training dataset has been contaminated with a set of impulse responses measured in a real apartment.

discussion with the Linguistic Data Consortium (LDC).

The reverberation time (T_{60}) of the considered room is about 0.7 seconds. Development and test data have been simulated with the same approach, but considering a different set of impulse responses.

The WSJ experiments are based on the popular wsj5k task (aligned with the CHiME 3 [10] task) and are conducted under two different acoustic conditions. For the *WSJ rev* case, the training set is contaminated with the same set of impulse responses adopted for TIMIT. For the *WSJ rev+noise* case, we also added non-stationary noises recorded in a domestic context (the average SNR is about 10 dB). The test phase is carried out with the DIRHA-English corpus (real-data part), consisting of 409 WSJ sentences uttered by six native American speakers in the above mentioned apartment. More details on this corpus and on the impulse responses adopted in this work can be found in [216, 226].

The features considered in this work are standard 39 Mel-Cepstral Coefficients (MFCCs) computed every 10 ms with a frame length of 25 ms. The speech enhancement DNNs are fed with a context of 21 consecutive frames and predict (every 10 ms) 11 consecutive frames of enhanced MFCC features. The speech recognition DNNs are fed by such 11 speech enhanced frames and predict both context-dependent and monophone targets at their output. All the layers used Rectified Linear Units (ReLU), except for the output of the speech enhancement DNNs (linear) and the output of the speech recognition modules (softmax). Batch normalization [123] and dropout [257] are employed for all the hidden layers. The labels for the speech enhancement DNN (denoted as x_{clean} in Figure 6.3) are the MFCC features of the original clean datasets. The labels for the speech recognition DNN (denoted as y_{SR} in Figure 6.3) are derived by performing a forced alignment procedure on the original training datasets. See the standard s5 recipe of Kaldi for more details [204].

The weights of the networks are initialized according to the *Glorot* ini-

tialization [89], while biases are initialized to zero. Training is based on a standard Stochastic Gradient Descend (SGD) optimization with mini-batches N of size 128. The performance on the development set is monitored after each epoch and the learning rate η is halved when the performance improvement is below a certain threshold. The training ends when no significant improvements have been observed for more than four consecutive epochs.

The main hyperparameters of the system (i.e., learning rate η, number of hidden layers, hidden neurons per layer, dropout factor, gradient weighting factor λ and number of unfolding levels L) have been optimized on the development set (DIRHA-English-WSJ5k set1 sim). As a result, speech enhancement and speech recognition DNNs with 4 hidden layers of 1024 neurons and DNNs with 6 hidden layers of 2048 neurons are employed for TIMIT and WSJ tasks, respectively. The initial learning rate is 0.08, the dropout factor is 0.2 and the considered number of levels is 3 ($l = 0, .., 2$). Similarly to [212], λ is fixed to 0.1.

The proposed system, which has been implemented with Theano [267], has been coupled with the Kaldi toolkit [204] to form a context-dependent DNN-HMM speech recognizer.