THE UNIVERSITY
of EDINBURGH

Learning Representations for Speech Recognition using Artificial Neural Networks

Paweł Świętojański

Doctor of Philosophy

Institute for Language, Cognition and Computation
School of Informatics
University of Edinburgh

2016

Lay Summary

Learning representations is a central challenge in machine learning. For speech recognition, which concerns mapping speech acoustics into sequences of words, we are interested in learning robust representations that are stable across different acoustic environments, recording equipment and irrelevant inter– and intra–speaker variabilities. This thesis is concerned with representation learning for acoustic model adaptation to speakers and environments, construction of acoustic models in low-resource settings, and learning representations from multiple acoustic channels. The investigations are primarily focused on the hybrid approach to acoustic modelling based on hidden Markov models and artificial neural networks (ANN).

The first contribution concerns acoustic model adaptation. This comprises two new adaptation transforms operating in ANN parameters space âĂŞ Learning Hidden Unit Contributions (LHUC) and differentiable pooling. Both operate at the level of activation functions and treat a trained ANN acoustic model as a canonical set of fixed-basis functions, from which one can later derive variants tailored to the specific distribution present in adaptation data. On average, depending on the characteristics of the test set, 5-25% relative increase in accuracies were obtained in an unsupervised adaptation setting (scenario in which human-level manual transcriptions are not available).

The second contribution concerns building acoustic models in low-resource data scenarios. In particular, we are concerned with insufficient amounts of transcribed acoustic material for estimating acoustic models in the target language. First we proposed an ANN with a structured output layer which models both context-dependent and context-independent speech units, with the context-independent predictions used at runtime to aid the prediction of context-dependent units. We also propose to perform multi-task adaptation with a structured output layer. Those propositions lead to consistent accuracy improvements for both low-resource speaker-independent acoustic modelling and adaptation with LHUC technique. We then demonstrate that one can build better acoustic models with unsupervised multi– and cross– lingual initialisation and find that pre-training is a largely language-independent. Up to 14.4% relative accuracy improvements are observed, depending on the amount of the available transcribed acoustic data in the target language.

The third contribution concerns building acoustic models from multi-channel acoustic data. For this purpose we investigate various ways of integrating and learning multi-channel representations. In particular, we investigate channel concatenation and the applicability of convolutional layers for this purpose. We propose a multi-channel convolutional layer with cross-channel pooling, which can be seen as a data-driven non-parametric auditory attention mechanism. We find that for unconstrained microphone arrays, our approach is able to match the performance of the comparable models trained on beamform-enhanced signals.

Abstract

Learning representations is a central challenge in machine learning. For speech recognition, we are interested in learning robust representations that are stable across different acoustic environments, recording equipment and irrelevant inter– and intra– speaker variabilities. This thesis is concerned with representation learning for acoustic model adaptation to speakers and environments, construction of acoustic models in low-resource settings, and learning representations from multiple acoustic channels. The investigations are primarily focused on the hybrid approach to acoustic modelling based on hidden Markov models and artificial neural networks (ANN).

The first contribution concerns acoustic model adaptation. This comprises two new adaptation transforms operating in ANN parameters space. Both operate at the level of activation functions and treat a trained ANN acoustic model as a canonical set of fixed-basis functions, from which one can later derive variants tailored to the specific distribution present in adaptation data. The first technique, termed Learning Hidden Unit Contributions (LHUC), depends on learning distribution-dependent linear combination coefficients for hidden units. This technique is then extended to altering groups of hidden units with parametric and differentiable pooling operators. We found the proposed adaptation techniques pose many desirable properties: they are relatively low-dimensional, do not overfit and can work in both a supervised and an unsupervised manner. For LHUC we also present extensions to speaker adaptive training and environment factorisation. On average, depending on the characteristics of the test set, 5-25% relative word error rate (WERR) reductions are obtained in an unsupervised two-pass adaptation setting.

The second contribution concerns building acoustic models in low-resource data scenarios. In particular, we are concerned with insufficient amounts of transcribed acoustic material for estimating acoustic models in the target language – thus assuming resources like lexicons or texts to estimate language models are available. First we proposed an ANN with a structured output layer which models both context–dependent and context–independent speech units, with the context-independent predictions used at runtime to aid the prediction of context-dependent states. We also propose to perform multi-task adaptation with a structured output layer. We obtain consistent WERR reductions up to

6.4% in low-resource speaker-independent acoustic modelling. Adapting those models in a multi-task manner with LHUC decreases WERRs by an additional 13.6%, compared to 12.7% for non multi-task LHUC. We then demonstrate that one can build better acoustic models with unsupervised multi– and cross– lingual initialisation and find that pre-training is a largely language-independent. Up to 14.4% WERR reductions are observed, depending on the amount of the available transcribed acoustic data in the target language.

The third contribution concerns building acoustic models from multi-channel acoustic data. For this purpose we investigate various ways of integrating and learning multi-channel representations. In particular, we investigate channel concatenation and the applicability of convolutional layers for this purpose. We propose a multi-channel convolutional layer with cross-channel pooling, which can be seen as a data-driven non-parametric auditory attention mechanism. We find that for unconstrained microphone arrays, our approach is able to match the performance of the comparable models trained on beamform-enhanced signals.

Acknowledgements

I would like to express my gratitude to:

- Steve Renals, for excellent supervision, patience, enthusiasm, scientific freedom and for the opportunity to pursue PhD studies in the Centre for Speech Technology Research (CSTR). It is hard to overstate how much I have benefited from Steve's knowledge, expertise and advice.

- My advisors: Peter Bell and Arnab Ghoshal, for engaging conversations and for many, much needed, suggestions during my PhD studies. Special thanks to Peter for proof-reading this dissertation.

- My examiners: Hervé Bourlard, Simon King and Andrew Senior for peer-reviewing this work and for the insightful comments which led to many improvements. All the shortcomings that remain are solely my fault.

- Catherine Lai, for proof-reading parts of this thesis and Jonathan Kilgour for offering help with the AMI data preparation. Special thanks to Thomas Hain for useful feedback on distant speech recognition experiments, and for sharing components of the SpandH AMI setup with us. Thanks to Iain Murray and Hiroshi Shimodaira for the feedback during my annual reviews.

- Everybody in the CSTR, for contributing towards a truly unique and superb academic environment.

- The NST project (special thanks to Steve Renals (again!), Mark Gales, Thomas Hain, Simon King and Phil Woodland for making it happen). There were many excellent researchers working in NST, interacting with whom was a very rewarding experience.

- Jinyu Li and Yifan Gong, for offering and hosting my two internships at Microsoft. Some of the ideas I developed there were used in this thesis.

- Finally, I would like to say thanks to my family, for the constant and unfailing support and to Anna for incredible patience, understanding and encouragement over the last several years.

This work was supported by EPSRC Programme Grant EP/I031022/1 *Natural Speech Technology* (NST). Thanks to the contributors of the Kaldi speech recognition toolkit and the Theano math expression compiler.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Paweł Świętojański*)

Table of Contents

Part I

Preliminaries

Chapter 1

Introduction

This thesis concerns *learning* and *adapting representations* for acoustic modelling in automatic speech recognition (ASR). We are primarily concerned with hidden Markov-based acoustic models [Baker, 1975] in which state probability distributions are estimated by connectionist architectures [Rosenblatt, 1962]. This is sometimes referred to as the *hybrid* approach to ASR first proposed by [Bourlard and Wellekens, 1990, Bourlard and Morgan, 1990], intensively developed in mid nineties [Bourlard et al., 1992, Renals et al., 1994, Bourlard and Morgan, 1994], and more recently refined to give state of the art results on multiple speech recognition benchmarks by Hinton et al. [2012] and many others.

Representation learning [Rumelhart et al., 1986] reflects a shift from a *knowledge-based* towards a *data-driven* means of extracting useful features from data. This process typically involves building multiple-levels of transformations jointly with a target predictor, driven by a single task-related objective. It is still possible to incorporate knowledge-motivated priors and assumptions; however, they tend to be much more generic when compared to the expertise engineered in hand-crafted features where one usually seeks for representation adjusted for strengths and weaknesses of the specific classifier.

Connectionist architectures have been very successful in representation learning in both *supervised* and *unsupervised* settings [Bengio et al., 2013]. This is not surprising and to a large degree may be attributed to gradient-based iterative learning algorithms [Rumelhart et al., 1986], which work well in practice. In addition, they provide modelling power which, under some mild assumptions, has been proved to be a *universal approximator* for smooth functions [Cybenko, 1989, Hornik et al., 1989, Barron, 1993]. Notice, however, that the universal approxi-

mation theorem says little about how to make use of those theoretical properties in practice or even how to relate the optimal model capacity to the given data. Nevertheless, ANNs offer an appealing unified framework for learning representations, allowing a focus on more generic aspects – for example, how to design an appropriate loss functions for the task or designing an optimiser capable of searching for model parameters which minimise this loss. At the same time ANNs give some guarantee that the model's parametrisation will have a sufficient capacity[1] to learn a satisfactory solution without worrying whether data fits predictor's capabilities. We will elaborate on this brief outline later in Chapter 2.

Despite significant progress in acoustic model development over the last several decades, including recent work on ANNs, ASR systems remain strongly domain dependent requiring specific task-oriented systems for good performance. Furthermore, it has been found that properly performed normalisation of the acoustic feature-space or direct adaptation of acoustic model parameters in a speaker-dependent manner yields significant gains in accuracy. This lack of generality is particularly pronounced for far-field speech recognition – in this case, the available corpora allow for controlled experiments marginalising the impact of all but the acoustic component on the final system performance to be carried out. The results show that extending the distance at which the speech is captured can double error rates – and this already assumes a matched scenario with two sets of independently trained models, one for distant and one for close-talking data.

This thesis is concerned with three major themes:

1. **Adapting representations:** We are primarily concerned with adaptation techniques operating in the parameter space of an ANN. In particular, we have developed a technique that relies on learning hidden unit contributions (`LHUC`) [Swietojanski and Renals, 2014] and offers a principled interpretation rooted in the universal approximation theorem. Building on `LHUC` we have extended it to a speaker-adaptive training variant able to work both in speaker independent and dependent manners [Swietojanski and Renals, 2016]. The work on `LHUC` took us towards related techniques that rely on differentiable pooling operators, in which adaptation is carried by estimating speaker-specific pooling regions [Swietojanski and Renals, 2015]. Within this avenue we focused on two types of pooling parametrisations - L_p and Gaussian weighted units. Both approaches were thoroughly

[1] In a sense that the number of model's parameters can be easily increased if necessary.

examined experimentally on multiple large vocabulary speech recognition (ASR) benchmarks. Results and broader motivations of those were reported in [Swietojanski, Li, and Renals, 2016] and [Swietojanski and Renals, 2016].

2. **Learning representations under low-resource acoustic conditions:** Adapting context-dependent models with small amounts of data results in very sparsely distributed adaptation targets – resulting in updating of only few outputs. To compensate for this we propose a multi-task adaptation with a structured output layer using an auxiliary context-independent layer to adapt context-dependent targets [Swietojanski, Bell, and Renals, 2015]. This structure also help in the generic scenario of speaker-independent low-source acoustic modelling.

   It is sometimes difficult to access transcribed data for some languages. As a result ASR systems in such scenarios offer much lower accuracy compared to similar systems estimated for resources-rich languages. Within this part we try to leverage multi-lingual acoustic data to improve ASR for a target under-resourced language. In particular, we propose unsupervised multi-lingual pre-training as a form of multi-lingual knowledge transfer [Swietojanski, Ghoshal, and Renals, 2012].

3. **Learning representations from multiple channels:** Much work has been done in multi-microphone acoustic modelling employing conventional Gaussian mixture models. The consensus found in the literature is that optimal performance involves beamforming of multiple-channels in order to create a single but enhanced audio signal. ANNs offer a very flexible framework to combine various sources of information and in Chapter 9 we investigate their applicability to model-based learning from multiple acoustic channels. In particular, we investigate different ways of combining multi-channel data in the model [Swietojanski, Ghoshal, and Renals, 2013a] and propose a convolutional layer with two-levels of pooling [Swietojanski, Ghoshal, and Renals, 2014a]. This approach, for unconstrained microphones, is able to match models trained on signal-enhanced speech [Renals and Swietojanski, 2014]. A side result of this work is the creation of publicly available recipes for the AMI and ICSI meeting corpora released with the Kaldi speech recognition toolkit (`www.kaldi-asr.org`).

1.1 Thesis Structure

Given above contributions, the thesis is organised as follows:

Chapter 2 provides a short introduction to connectionist models.

Chapter 3 contains a brief and high level review of large vocabulary speech recognition.

Chapter 4 gathers together descriptions and statistics of the corpora used to carry out the experiments. We follow an in-place approach in this work – we present experimental results in each chapter following theoretical descriptions.

Chapter 5 reviews adaptation methods and introduces Learning Hidden Unit Contributions (LHUC) and its speaker adaptively trained (SAT) variant.

Chapter 6 continues the adaptation topic and presents model-based adaptation using parametric and differentiable pooling operators.

Chapter 7 investigates low-resource acoustic models and adaptation using a multi-task training and adaptation with structured output layer.

Chapter 8 concerns unsupervised multi-lingual knowledge transfer.

Chapter 9 investigates the use of multiple channels in distant speech recognition, where our primary focus lies in explicit representation learning from multi-channel acoustic data.

Chapter 10 contains a summary and discusses possible future work.

1.2 Declaration of content

This thesis is almost completely composed of the work published in the following journal and conference papers:

- P. Swietojanski and S. Renals. Differentiable Pooling for Unsupervised Acoustic Model Adaptation. IEEE/ACM Transactions on Audio, Speech and Language Processing, 2016

- P. Swietojanski J. Li and S. Renals. Learning Hidden Unit Contributions for Unsupervised Acoustic Model Adaptation. IEEE/ACM Transactions on Audio, Speech and Language Processing, 2016

- P. Swietojanski and S. Renals. SAT-LHUC: Speaker Adaptive Training for Learning Hidden Unit Contributions. In Proc. IEEE International Con-

ference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China, 2016

- P. Swietojanski, P. Bell, and S. Renals. Structured output layer with auxiliary targets for context-dependent acoustic modelling. In Proc. ISCA Interspeech, Dresden, Germany, 2015.

- P. Swietojanski and S. Renals. Differentiable pooling for unsupervised speaker adaptation. In Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brisbane, Australia, 2015

- P. Swietojanski and S. Renals. Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models. In Proc. IEEE Spoken Language Technology Workshop (SLT), Lake Tahoe, USA, 2014

- P. Swietojanski, A. Ghoshal, and S. Renals. Convolutional neural networks for distant speech recognition. IEEE Signal Processing Letters, 21(9):1120-1124, September 2014

- S. Renals and P. Swietojanski. Neural networks for distant speech recognition. In The 4th Joint Workshop on Hands-free Speech Communication and Microphone Arrays (HSCMA), Nancy, France, 2014

- P. Swietojanski, A. Ghoshal, and S. Renals. Hybrid acoustic models for distant and multichannel large vocabulary speech recognition. In Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), Olomouc, Czech Republic, 2013

- P. Swietojanski, A. Ghoshal, and S. Renals. Unsupervised cross-lingual knowledge transfer in DNN-based LVCSR. In Proc. IEEE Spoken Language Technology Workshop (SLT), pages 246-251, Miami, Florida, USA, 2012.

Some related aspects in the thesis were inspired, but did not get into directly, by the following work I have (co-)authored:

- P. Swietojanski, J-T Huang and J. Li. Investigation of maxout networks for speech recognition. In Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 2014

- P. Swietojanski, A. Ghoshal, and S. Renals. Revisiting hybrid and GMM-HMM system combination techniques. In Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Vancouver, Canada, 2013

- P. Bell, P. Swietojanski, and S. Renals. Multi-level adaptive networks in tandem and hybrid ASR systems. In Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Vancouver, Canada, 2013

- A. Ghoshal, P. Swietojanski, and S. Renals. Multilingual training of deep neural networks. In Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Vancouver, Canada, 2013

Beyond above works, during my PhD days I also had a chance to contribute towards related research projects and speech recognition evaluations (IWSLT12, IWSLT14 and MGB Challenge 2015), some of those collaborations resulted in peer-reviewed publications, as follows:

- Z. Wu, P. Swietojanski, C. Veaux, S. Renals, and S. King. A study of speaker adaptation for DNN-based speech synthesis. In Proc. ISCA Interspeech, Dresden, Germany, 2015

- P. Bell, P. Swietojanski, J. Driesen, Mark Sinclair, Fergus McInnes, and Steve Renals. The UEDIN ASR systems for the IWSLT 2014 evaluation. In Proc. International Workshop on Spoken Language Translation (IWSLT), South Lake Tahoe, USA, 2014

- P. Bell, H. Yamamoto, P. Swietojanski, Y. Wu, F. McInnes, C. Hori, and S. Renals. A lecture transcription system combining neural network acoustic and language models. In Proc. ISCA Interspeech, Lyon, France, 2013

- H. Christensen, M. Aniol, P. Bell, P. Green, T. Hain, S. King, and P. Swietojanski. Combining in-domain and out-of-domain speech data for automatic recognition of disordered speech. In Proc. ISCA Interspeech, Lyon, France, 2013

- P. Lanchantin, P. Bell, M. Gales, T. Hain, X. Liu, Y. Long, J. Quinnell, S. Renals, O. Saz, M. Seigel, P. Swietojanski, and P. Woodland. Automatic

transcription of multi-genre media archives. In Proc. Workshop on Speech, Language and Audio in Multimedia, Marseille, France, 2013

- P. Bell, M. Gales, P. Lanchantin, X. Liu, Y. Long, S. Renals, P. Swietojanski, and P. Woodland. Transcription of multi-genre media archives using out-of-domain data. In Proc. IEEE Spoken Language Technology Workshop (SLT), pages 324-329, Miami, Florida, USA, 2012

- E. Hasler, P. Bell, A. Ghoshal, B. Haddow, P. Koehn, F. McInnes, S. Renals, and P. Swietojanski. The UEDIN system for the IWSLT 2012 evaluation. In Proc. International Workshop on Spoken Language Translation (IWSLT), Hong Kong, China, 2012

1.3 Notation

The following notation has been used throughout this work.

$\boldsymbol{o}_t$	speech observation at time t
$\boldsymbol{o}_t^{\mathsf{r}}$	the "static" elements of feature vector $\boldsymbol{o}_t$
$\Delta\boldsymbol{o}_t^{\mathsf{r}}$	the "delta" elements of feature vector $\boldsymbol{o}_t$
$\Delta^2\boldsymbol{o}_t^{\mathsf{r}}$	the "delta-delta" elements of feature vector $\boldsymbol{o}_t$
$\bar{\boldsymbol{O}}_t$	context window of $2c+1$ speech observations: $\bar{\boldsymbol{O}}_t = [\boldsymbol{o}_{t-c}^\top, \ldots, \boldsymbol{o}_t^\top, \ldots, \boldsymbol{o}_{t+c}^\top]^\top$
$\boldsymbol{O}_{1:T}$	observation sequence $\boldsymbol{o}_1, \ldots, \boldsymbol{o}_T$
$\boldsymbol{\theta}$	an arbitrary set of parameters
$\mathcal{F}(\boldsymbol{\theta})$	a criterion to be optimised over parameters $\boldsymbol{\theta}$
$f(\mathbf{x};\boldsymbol{\theta})$	an arbitrary neural network, parametrised by $\boldsymbol{\theta}$, acting on input $\mathbf{x}$
$f^l(\mathbf{x};\boldsymbol{\theta}^{\mathbf{l}})$	lth layer of neural network, parametrised by $\boldsymbol{\theta}^l$
$\mathbf{W}^l$	weight matrix associated with layer l of a neural network
$\mathbf{w}_i^l$	weight vector associated with node i for layer l
$\mathbf{b}^l$	bias vector at lth layer
$\phi^l(\cdot)$	non-linear function applied at lth layer
$\mathbf{h}^l$	vector of hidden activations at lth layer
c_m	the prior of Gaussian component m
$\boldsymbol{\mu}^{(m)}$	the mean of Gaussian component m

$\boldsymbol{\Sigma}^{(m)}$	the covariance matrix of Gaussian component m
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$
$\mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}, \boldsymbol{\Sigma})$	likelihood of observation $\boldsymbol{o}_t$ being generated by $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$
$p(\boldsymbol{O}_{1:T}; \boldsymbol{\theta})$	likelihood of observation sequence $\boldsymbol{O}_{1:T}$ given parameters $\boldsymbol{\theta}$
q_j	state j of the HMM
$\mathbf{Q}$	the HMM state sequence $q_{1:T}$
$\mathcal{Q}$	set of all HMM states
$\mathbf{A}$	an arbitrary transformation matrix
$\mathbf{A}^\top$	transpose of $\mathbf{A}$
$\mathbf{A}^{-1}$	inverse of $\mathbf{A}$
$\lvert\mathbf{A}\rvert$	determinant of $\mathbf{A}$
a_{ij}	element i, j of $\mathbf{A}$

Acronyms

ANN	artificial neural network
BMMI	boosted maximum mutual information
ASR	automatic speech recognition
CMN	cepstral mean normalisation
CMLLR	constrained maximum likelihood linear regression
CNN	ANN with at least one convolutional layer
CVN	cepstral variance normalisation
DNN	(deep) artificial neural network (the same as ANN)
EM	expectation maximisation
EBP	error back-propagation
FBANK	log mel filterbank
FFT	fast Fourier transform
GMM	Gaussian mixture model
HMM	hidden Markov model
LVCSR	large vocabulary continuous speech recognition
LDA	linear discriminant analysis

LHUC	learning hidden unit contributions
LM	language model
MBR	minimum Bayes risk
MFCC	mel-frequency cepstral coefficients
ML	maximum likelihood
MLLR	maximum likelihood linear regression
MLLT	maximum likelihood linear transform
MMI	maximum mutual information
MMSE	minimum mean square error
PLP	perceptual linear prediction
PPL	perplexity of a language model
PDF	probability density function
RBM	restricted Boltzmann machine
RNN	recurrent neural network
SAT	speaker adaptive training
SOL	structured output layer
SD	speaker dependent parameters
SI	speaker independent parameters
SNR	signal to noise ratio
TDOA	time delay of arrival
VTLN	vocal tract length normalisation
WER	word error rate

Chapter 2

Connectionist Models

This chapter presents the background on the construction and learning of artificial neural network models.

2.1 Introduction

The concept of connectionism dates back as far as 400 B.C. and Aristotle's notion of mental association [Medler, 1998] and has been since influenced by many fields of science and philosophy, as outlined in the survey by Medler [1998]. In this thesis we are interested in the "post-perceptron" era of connectionism, or an *engineering* approach, especially in the context of the distributed parallel processing framework proposed by [Rumelhart et al., 1986] and the definition given in Bourlard and Morgan [1994] which describes a connectionist system as a: *System in which information is represented and processed in terms of the input pattern and strength of connections between units that do some simple processing on their input.*

In the literature, connectionist systems are often referred to as {artificial, multi-layer, deep} neural networks or multi-layer perceptrons. Hereafter in this thesis we will primarily rely on feed-forward structures and we will use term artificial neural network (ANN) when referring to such models, where necessary we will also explicitly specify the type of connectivity pattern of its parameters (feed-forward, recurrent, convolutional, or a combination of those).

2.2 ANNs as a computational network

A convenient way to explain the computations performed by an arbitrary connectionist model is by treating the computation as traversing and executing a series of computational nodes organised in a graph [LeCun et al., 1998a]. Denote by $\mathbb{G} = \{\mathbb{V}, \mathbb{E}\}$ a graph consisting of a set of vertices (or nodes) $\mathbb{V}$ and a set of connecting edges (or links) $\mathbb{E}$. Such a graph must exhibit certain properties: i) its edges should be *directed*, thus defining one-directional dependencies between vertices $\mathbb{V}$ and ii) paths defined by edges (possibly spanning many vertices) must not create *cycles*, given a specified dependency structure. This is a non-rigorous definition of a *Directed Acyclic Graph* (DAG).

Figure 2.1 (a) depicts an illustration of such a DAG with four vertices $\mathbb{V} = \{A, B, C, D\}$, each of those vertices as well as the whole block computes a function $\mathbf{Y}$ from inputs $\mathbf{X}$ using parameters $\boldsymbol{\theta}$. The dependency structure allows us to easily find the order in which one need to execute the operations – this is known as a *topological ordering* and can be obtained by traversing the graph in depth-first order, starting from its inputs (root). Assuming we know how to perform computations at each node, following this topological ordering and evaluating visited nodes will effectively perform the forward-pass procedure. In a similar way, starting from the forward computation DAG, one can build a similar DAG whose traversal will result in the error back-propagation procedure [Rumelhart et al., 1986], as illustrated in Figure 2.1 (b).

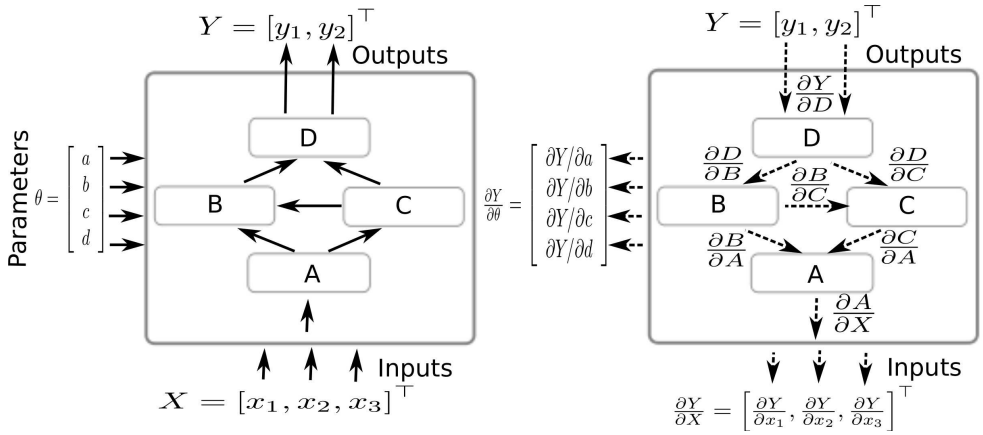


Figure 2.1: (Left) Computational graph consisting of some 4 internal computations, A, B, C and D parametrised by a, b, c and d, respectively. (Right) corresponding back-propagation graph

Typically, each computational node is required to implement three types of computations:

- Forward computation through the node,
- Backward computation: gradient of the outputs of the node with respect to its inputs,
- Gradient of the outputs of the node with respect to its parameters (if any)

The loss function, used to guide the learning process, can also be represented as a node with two inputs (targets, model predictions) and a scalar output representing cost which allows DAG representations of arbitrary learnable ANN structures, as long as each component knows how to perform the forward and backward computations.

Examples presented hitherto underline some generic principles allowing us to build arbitrarily complex computational graphs keeping the underlying computational machinery the same. In the next section, we outline some of the typical choices for such nodes, divided into three categories: loss functions, linear transforms and activation functions.

2.3 ANN components

A feed-forward ANN is implemented as a nested function comprising of L processing layers, parametrised by $\boldsymbol{\theta}$, which maps some input $\mathbf{x}$ into output $\mathbf{y}$:

$$f(\mathbf{x};\boldsymbol{\theta}) = f^L\left(f^{L-1}\left(\ldots f^2\left(f^1\left(\mathbf{x};\boldsymbol{\theta}^1\right);\boldsymbol{\theta}^2\right)\ldots;\boldsymbol{\theta}^{L-1}\right);\boldsymbol{\theta}^L\right) \tag{2.1}$$

The particular mapping that an ANN can learn will depend on the type of connectivity pattern and hidden activation function in each layer, and the type of output layer activation and the loss function. We review them briefly below.

2.3.1 Loss functions

In order to learn the parameters of an ANN, we must specify a suitable loss function. The loss function, to be minimised, does not necessarily need to be easy to evaluate, as long as one can compute its gradient or provide a satisfactory approximation to it. Below we list some of the popular choices that often also

act as surrogate loss functions for the tasks in which more suitable losses are not available. Let $\mathbf{y}$ be the output layer activations produced by a neural network $f(\mathbf{x};\boldsymbol{\theta})$ and let $\mathbf{t}$ denote training targets for a given task, then we have:

- Squared Errors: A suitable loss for regression tasks when the target $\mathbf{t}$ contain real numbers and $f(\mathbf{x};\boldsymbol{\theta})$ produces linear activations:

$$\mathcal{F}_{SE}(\mathbf{t},\mathbf{y}) = \frac{1}{2}||\mathbf{t}-\mathbf{y}||^2 \tag{2.2}$$

$$\frac{\partial \mathcal{F}_{SE}(\mathbf{t},\mathbf{y})}{\partial \mathbf{y}} = \mathbf{y}-\mathbf{t} \tag{2.3}$$

- Cross-entropy: in cases where both $\mathbf{t}$ and $\mathbf{y}$ are categorical probability distributions (i.e. for any valid output i, $0 < y_i < 1$ and $\sum_i y_i = 1$). It is suitable to minimise the Kullback-Leibler divergence of $\mathbf{y}$ from $\mathbf{t}$:

$$KL(\mathbf{t}||\mathbf{y}) = \sum_i t_i \log \frac{t_i}{y_i} \tag{2.4}$$

$$= \sum_i t_i \log t_i - \sum_i t_i \log y_i \tag{2.5}$$

  Since the parameters $\boldsymbol{\theta}$ of an ANN $f(\mathbf{x};\boldsymbol{\theta})$ used to produce $\mathbf{y}$ do not depend on the first part of (2.5) the loss to minimise, and its gradient with respect to $\mathbf{y}$ are:

$$\mathcal{F}_{CE}(\mathbf{t},\mathbf{y}) = -\sum_i t_i \log y_i \tag{2.6}$$

$$\frac{\partial \mathcal{F}_{CE}(t_i,y_i)}{\partial y_i} = -\frac{t_i}{y_i} \tag{2.7}$$

- Binary cross-entropy is a special case of cross-entropy when there are only two classes or in scenarios in which activations at the output layer are probabilistic but independent from each other, in which case the loss and its gradient with respect to y is:

$$\mathcal{F}_{BCE}(t,y) = -t\log y - (1-t)\log(1-y) \tag{2.8}$$

$$\frac{\partial \mathcal{F}_{CE}(t,y)}{\partial y} = -\frac{t}{y} + \frac{1-t}{1-y} \tag{2.9}$$

- Other: Ideally one would like to optimise the loss which directly corresponds to the performance metric of the task at hand. For example in speech transcription, where one is interested in optimising sequential performance, customised loss functions may yield improved performance [Povey, 2003, Kingsbury, 2009] (see also Section 3.3.4).

2.3.2 Linear transformations

In this section we describe two types of linear transformations commonly used with ANN in fully-connected and convolutional settings.

- Affine transform: implements a dot product and a shift:

$$\mathbf{a} = \mathbf{W}^\top \mathbf{x} + \mathbf{b} \tag{2.10}$$

by taking a derivatives w.r.t the input ($\mathbf{x}$) and the transform parameters ($\boldsymbol{\theta} = \{\mathbf{W}, \mathbf{b}\}$) we have:

$$\frac{\partial \mathbf{a}}{\partial \mathbf{x}} = \mathbf{W}, \quad \frac{\partial \mathbf{a}}{\partial \mathbf{W}} = \mathbf{x}, \quad \frac{\partial \mathbf{a}}{\partial \mathbf{b}} = \mathbf{1} \tag{2.11}$$

- Affine transform with convolution: implements a local connectivity pattern which is typically shared across multiple locations in the layer's input space [LeCun and Bengio, 1995]. Typically, with ANNs we perform convolution along one, two or three dimensions. In particular, in this thesis we use one dimensional convolution along the frequency axis as in [Abdel-Hamid et al., 2014], in which case the k-th convolutional filter (out of total J such filters in a layer) can be expressed as a dot product[1]. Denote by $\mathbf{x} = [x_1, x_2, \ldots, x_M]^\top$ some M-dimensional input signal and by $\mathbf{w}^k = [w_1^k, w_2^k, \ldots, w_N^k]^\top$ the weights of the kth N-dimensional convolutional filter. Then one can build a large but sparse matrix $\mathbf{W}^k$ by replicating $\mathbf{w}^k$ filters as in (2.12) and then computing the convolutional linear activations for each of J filters as in (2.10).

$$\mathbf{W}^{k\top} = \begin{bmatrix} w_1^k & w_2^k & \ldots & w_N^k & 0 & 0 & \ldots & 0 \\ 0 & w_1^k & w_2^k & \ldots & w_N^k & 0 & \ldots & 0 \\ 0 & 0 & w_1^k & w_2^k & \ldots & w_N^k & \ldots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \ldots & 0 & w_1^k & w_2^k & \ldots & w_N^k \end{bmatrix} \tag{2.12}$$

Extra steps must be taken when computing the gradients with respect to parameters, in which case one needs to take into account the fact the kth kernel $\mathbf{w}^k$ is shared across many locations, and the total gradient is the sum of the intermediate gradients computed with $\mathbf{w}^k$ at multiple locations in $\mathbf{x}$.

[1] One does not need to flip the filter index (in order to perform convolution in the mathematical sense) as the filters are data-driven and the same result can be obtained using *cross-correlation* instead, which is more efficient computationally.

2.3.3 Activation functions

Linear transformations are usually followed by non-linear activation functions. Those functions can either act element-wise or take into account groups of units. Assuming $\mathbf{a}$ is the vector of linear activations obtained with (2.10) and a_i is its ith element, popular choices for non-linear functions are as follows:

- Identity: results in no transformation of linear activations in (2.10). These are typically used in the output layer for regression tasks, but also at hidden layers with bottle-neck feature extractors and as an adaptation transform.

$$\phi(a_i) = a_i \tag{2.13}$$

- Sigmoid: squashing non-linearity resulting in activations bounded by $[0, 1]$:

$$\phi(a_i) = \frac{1}{1 + \exp(-a_i)} \tag{2.14}$$

$$\frac{\partial \phi(a_i)}{\partial a_i} = \phi(a_i)(1 - \phi(a_i)) \tag{2.15}$$

- Hyperbolic tangent (tanh): squashing non-linearity resulting in activations bounded by $[-1, 1]$:

$$\phi(a_i) = \frac{\exp(a_i) - \exp(-a_i)}{\exp(a_i) + \exp(-a_i)} \tag{2.16}$$

$$\frac{\partial \phi(a_i)}{\partial a_i} = 1 - \phi(a_i)^2 \tag{2.17}$$

- Rectified Linear Unit (ReLU) [Nair and Hinton, 2010]: ReLU implements a form of data-driven sparsity – on average the activations are sparse, but the general sparsity pattern will depend on particular data-point. This is different from the sparsity obtained in parameters space with L_1 regularisation (see also Section 2.5) as the latter affects all data-points in the same way. Given a linear activation a_i, ReLU forward and backward computations are given as follows:

$$\phi(a_i) = \max(0, a_i), \tag{2.18}$$

$$\frac{\partial \phi(a_i)}{\partial a_i} = \begin{cases} 1 & \text{if } a_i > 0 \\ 0 & \text{if } a_i < 0 \end{cases} \tag{2.19}$$

- Maxout (or max-pooling in general) [Riesenhuber and Poggio, 1999, Goodfellow et al., 2013]: maxout is an example of a type of data-driven non-linearity in which the transfer function can be learned from data – the model can build a non-linear activation from piecewise linear components. These linear components, depending on the number of linear regions used in the pooling operator, can approximate other functions, such as ReLU or absolute value. Given some set $\{a_i\}_{i \in R}$ of R linear activations, maxout implements the following operation:

$$\phi(\{a_i\}_{i \in R}) = \max(\{a_i\}_{i \in R}) \tag{2.20}$$

$$\frac{\partial \phi(\{a_i\}_{i \in R})}{\partial a_{i \in R}} = \begin{cases} 1 & \text{if } a_i \text{ is the max activation} \\ 0 & \text{otherwise} \end{cases} \tag{2.21}$$

- L_p–norm [Boureau et al., 2010]: the activation is implemented as an L-norm with an arbitrary order $p \geq 1$ (in order to satisfy triangle inequality, see also Chapter 6) over a set $\{a_i\}_{i \in R}$ of R linear activations:

$$\phi(\{a_i\}_{i \in R}) = \left(\sum_{i \in R} |a_i|^p \right)^{\frac{1}{p}} \tag{2.22}$$

$$\frac{\partial \phi(\{a_i\}_{i \in R})}{\partial a_{i \in R}} = \frac{a_i \cdot |a_i|^{p-2}}{\sum_{j \in R} |a_j|^p} \cdot \phi(\{a_i\}_{i \in R}) \tag{2.23}$$

- Softmax [Bridle, 1990]: To obtain a probability distribution one can use the softmax non-linearity (2.24) which is a generalisation of sigmoid activations to multi-class problems. Softmax operation is prone to both over– and under–flows. Over-flows are typically addressed by subtracting the maximum activation $m = \max(\mathbf{a})$ before evaluating exponentials, as in the rightmost part of (2.24). If necessary, under–flows can be avoided with applying the same trick and computing the logarithm of softmax instead.

$$\phi(a_i) = \frac{\exp(a_i)}{\sum_j \exp(a_j)} = \frac{\exp(a_i - m)}{\sum_j \exp(a_j - m)} \tag{2.24}$$

  The derivative of softmax with respect to linear activation is given in (2.25), where δ_{ij} is the Kronecker delta function:

$$\frac{\phi(a_i)}{a_j} = \phi(a_i)(\delta_{ij} - \phi(a_j)) \tag{2.25}$$

2.4 Training

Most techniques for estimating ANN parameters are gradient-based methods operating in either online or batch modes. The dominant technique uses stochastic gradient descent (SGD) [Bishop, 2006], possibly in a parallelised variant [Dean et al., 2012, Povey et al., 2014]. Second order methods have also been reported to work well with ANN [Martens, 2010, Sainath et al., 2013b]. The potential advantage of second order methods (in return for higher computational load) lies in taking into account the quadratic approximation to the weight-space curvature, (theoretically) easier parallelisation and no requirement to tune learning rates (though usually other hyper-parameters are introduced in exchange). Experience shows that well tuned SGD algorithm offers results comparable to second order optimisers [Sutskever et al., 2013].

The ANN models in this work were trained with stochastic gradient descent, which for a neural network of the form $\mathbf{y} = f(\mathbf{x}; \boldsymbol{\theta})$ can be summarised as follows:

1. Randomly initialise model parameters $\boldsymbol{\theta}$, set momentum parameters $\boldsymbol{\vartheta}$ to 0
2. Until stopping condition has been satisfied
3. Draw B examples (a mini-batch) from training data: $(\mathbf{x}_i, t_i)$ for $i = 1, \ldots, B$
4. Perform a forward pass to get predictions $\mathbf{y}$ for the mini-batch
5. Compute the mean loss $\mathcal{F}(\mathbf{t}, \mathbf{y})$ and the gradient with respect to $\mathbf{y}$
6. Back-propagate the top-level errors $\partial\mathcal{F}(\mathbf{t}, \mathbf{y})/\partial\mathbf{y}$ down through the network using multi-path multi-chain rule
7. For each learnable parameter $\boldsymbol{\theta}_j \in \boldsymbol{\theta}$ compute its gradient $\boldsymbol{g}(\boldsymbol{\theta}_j)$ on B examples in the mini-batch:
$$\boldsymbol{g}(\boldsymbol{\theta}_j) = \frac{1}{B} \sum_{i}^{B} \frac{\partial}{\partial \boldsymbol{\theta}_j} \mathcal{F}(\mathbf{x}_i; \boldsymbol{\theta}_j) \tag{2.26}$$
8. Update the current model parameters with the newly computed gradients, $\alpha \in [0, 1]$ is the momentum coefficient and $\epsilon > 0$ denotes a learning rate:
$$\boldsymbol{\vartheta}_j^{t+1} = \alpha \boldsymbol{\vartheta}_j^t - \epsilon \boldsymbol{g}(\boldsymbol{\theta}_j) \tag{2.27}$$
$$\boldsymbol{\theta}_j^{t+1} = \boldsymbol{\theta}_j^t + \boldsymbol{\vartheta}_j^{t+1} \tag{2.28}$$
9. If there are more mini-batches to process go to step 3 otherwise go to step 2

2.5 Regularisation

Optimising loss functions mentioned in Section 2.3 from limited training data results in ANN models that may poorly generalise to unseen conditions, especially when the model is powerful enough to memorize spurious specifics in the training examples. To account for this one usually applies some *regularisation* terms on the ANN parameters to prevent over-fitting.

Typically, regularisation adds a complexity term to the cost function. Its purpose is to put some prior on the model parameters to prevent them from learning undesired configurations. The most common priors assume smoother solutions (ones which are not able to fit training data too well) are better as they are more likely to better generalise to unseen data.

A way to incorporate such a prior in the model is to add a complexity term to the training criterion in order to penalise certain configurations of the parameters – either from growing too large with weight decay (L_2 penalty) or preferring a solution that can be modelled with fewer parameters (L_1), hence encouraging sparsity. An arbitrary objective with L_1 and L_2 penalty has the following form, with $\lambda_1 \geq 0$ and $\lambda_2 \geq 0$ some small positive constants usually optimised on a development set.

$$\mathcal{F}_{reg}(\boldsymbol{\theta}) = \mathcal{F}_{train}(\boldsymbol{\theta}) + \lambda_1|\boldsymbol{\theta}| + \lambda_2||\boldsymbol{\theta}||^2 \tag{2.29}$$

There exist other effective regularisation techniques, for example, *dropout* [Srivastava et al., 2014] – a technique that omits a specified percentage of random hidden units (or inputs) during training or pre-training with stacked restricted Boltzmann machines [Hinton et al., 2006] for which one the explanations for its effectiveness in initialising ANN parameters is rooted in regularisation [Erhan et al., 2010].

2.6 Summary

We have briefly reviewed the basics behind a machinery of building and training artificial neural networks, including a brief overview of computational networks, loss functions, linear transformations and activation functions. We also outlined stochastic gradient descent and described the basics of regularisation.

Chapter 3

Overview of Automatic Speech Recognition

This chapter gives a high level overview of hidden Markov model-based speech recognition systems. More detailed comments have been given in the context of the material that is of particular interest to the follow up of this thesis, or to developments that were of seminal importance for speech recognition progress. Note however, this view may be in places biased by my own personal perspective.

3.1 Brief historical outlook

Automatic Speech Recognition (ASR) has a long history with work in machine-based classification of isolated speech units dating back to the 1950s [Davis et al., 1952]. To add some perspective to this picture, Alan Turing published his seminal work on computable functions in [Turing, 1936] and the first practical implementation of a Turing-complete machine is considered to be ENIAC (Electronic Numerical Integrator and Computer), built 9 years later in 1945. The quest to develop artificial intelligence (AI) has been around for much longer, though it was not clear how AI might be evaluated until another seminal work of Turing [1950] and the concept of *imitation game*, known today as a *Turing test*. Arguably, human-level artificial intelligence requires a natural-language processing component. The system of [Davis et al., 1952] did not use generic computing machines nor notions of what we consider nowadays as machine learning [Mitchell, 1997]; rather it was implemented as a specialised end-to-end circuit designed to extract, match and announce the recognised digit patterns found in speech uttered by a

single speaker – for whom the recognition circuit needed to be deliberately tuned prior to usage. This tuning, however, did not involve any form of automatic learning but rather relied on a manual incorporation of the necessary knowledge about unseen speaker's acoustic characteristics by a human expert. In a sense though, the system had some traits of what we consider nowadays as a standard pattern recognition pipeline involving stages like *feature extraction* and *matching* those features against some pre-programmed templates.

Unsurprisingly, a step forward in terms of creating more generic ASR systems was the invention of more sophisticated classifiers and better ways of extracting features from acoustic signals. An example of the former is the dynamic time warping (DTW) algorithm [Vintsyuk, 1968], a form of nearest neighbour classifier [Fix and Hodges, 1951], designed to find a similarity score between known and unseen sequences with different durations. This approach initially had some success in recognition of isolated [Velichko and Zagoruyko, 1970] as well as connected [Sakoe and Chiba, 1978] words, from closed and rather limited vocabularies (up to several hundreds words). Progress in extracting better speech features from the acoustic signal was stimulated by the invention of efficient algorithms for time-frequency domain transformations, in particular a fast implementation of the Fourier transform by Cooley and Tukey [1965], which enabled wide investigation of knowledge-motivated spectral feature extractors [Stevens et al., 1937], for example, mel-frequency cepstral coefficients [Mermelstein, 1976]. The DTW algorithm operating on frequency-derived acoustic features still remains the method of choice for tasks in which an additional domain knowledge about acoustic and language structure is severely limited, or at least as a building block of more sophisticated systems, in particular as a form of similarity measure in unsupervised approaches to word discovery. DTW is an example of a *non-parametric* approach and as such does not explicitly learn any statistical regularities or a representation of the problem from the training data. In this thesis, we will be primarily concerned with *parametric* families of models, that is the models that can extract and incorporate some task-related knowledge using their parameterisation, which can have either statistical or deterministic interpretations. In parametric approaches, once the process of estimating model parameters is finished, one no longer needs the training examples, unlike non-parametric template matching systems.

The framework that forms much of the foundations of current ASR systems emerged during 1970s from the work of [Baker, 1975] and [Jelinek, 1976]. The

transformative idea was to cast the speech to text transcription problem as a hierarchical generative process using hidden Markov models (HMM) where acoustic events are treated as random variables and depend probabilistically on the internal (hidden) system *states*. This approach allowed parameterisation of the speech generation model by two sets of conditional probability distributions. It suffices to say here that hierarchies of HMMs can represent units of phonemes, words or sentences, and the conditional probabilities of interest can be reliably estimated from representative speech training corpora. Initially HMM-based systems utilised discrete distributions. These were replaced by continuous density multivariate Gaussian mixture models (GMM) proposed by [Rabiner et al., 1985]. GMMs quickly gained popularity and many techniques addressing their early weaknesses have been proposed - some of them will be in more detail described later. An excellent introduction to early GMM-HMM systems can be found in [Rabiner, 1989] and [Gales and Young, 2008] provide an overview of many later improvements.

3.2 Introduction to data-driven speech recognition

Speech recognition, or *speech to text transcription*, is an example of a *sequence to sequence* mapping problem where the input consists of a sequence of acoustic features $\mathbf{O} = [\mathbf{o}_1, \ldots, \mathbf{o}_T]$ and one is interested in finding the underlying sequence of words $\boldsymbol{w} = [w_1, \ldots, w_K]$, with $K << T$. In a statistical framework, which is by far the most successful approach to date, one usually defines the ASR task in terms of finding the word sequence $\boldsymbol{w}^*$ out of all possible hypotheses $\mathcal{H}$ which maximises the posterior probability given the acoustic observations $\mathbf{O}$:

$$\boldsymbol{w}^* = \arg\max_{\boldsymbol{w} \in \mathcal{H}} P(\boldsymbol{w}|\mathbf{O}) \tag{3.1}$$

Historically, building models solving $P(\boldsymbol{w}|\mathbf{O})$ directly proved to be difficult[1] and the problem is typically decomposed using Bayes' rule into *likelihood* $p(\mathbf{O}|\boldsymbol{w})$,

[1]Note there have been attempts to model this posterior directly, for example using conditional random fields [Gunawardana et al., 2005, Zweig and Nguyen, 2009] or recurrent neural networks [Graves and Jaitly, 2014, Chan et al., 2015, Lu and Renals, 2016]. Neither of those methods (at this stage) provide competitive results on large vocabulary speech recognition and will not be considered in this thesis. Also notice, handling $P(\boldsymbol{w}|\mathbf{O})$ explicitly is different from altering the generative model's parameters of $p(\mathbf{O}|\boldsymbol{w})$ in a sequence discriminative manner, as described later.

prior $P(\boldsymbol{w})$ and normalisation $p(\mathbf{O})$ terms, as follows:

$$\boldsymbol{w}^* = \arg\max_{\boldsymbol{w}\in\mathcal{H}} \frac{p(\mathbf{O}|\boldsymbol{w})P(\boldsymbol{w})}{p(\mathbf{O})} \tag{3.2}$$

$$= \arg\max_{\boldsymbol{w}\in\mathcal{H}} p(\mathbf{O}|\boldsymbol{w})P(\boldsymbol{w}) \tag{3.3}$$

The marginal probability $p(\mathbf{O})$ in (3.2) is independent of the hypothesised word sequence and is ignored during decoding, thus implementing (3.3). $p(\mathbf{O}|\boldsymbol{w})$ and $P(\boldsymbol{w})$ are referred to as the *acoustic model* and the *language model*, and are estimated independently, often using different corpora.

A simplified illustration of an ASR system and analysis of an example phrase is depicted in Figure 3.1. Block (a) of Figure 3.1 presents a time-domain representation of a speech signal which was transcribed at the word-level and will be treated as a *reference transcription* for training purposes.

As shown in block (b) of Figure 3.1 this word-level transcription is transformed into a phonetic-level transcription using a *lexicon* which maps words to their phonetic pronunciations. This illustration assumes that HMM models are built for context-independent phonemes[2], as visualised in the middle bar of the block (b). The sequence of those composite models, which happen to be HMMs (formally introduced in Section 3.3) also determines the sequence of the underlying states $\mathbf{Q}_{1:T} = [q_1, \ldots, q_T]$, given the reference utterance $\boldsymbol{w}^{ref} = [\text{Transcribe}, \text{me}, \text{now}]$.

Block (c) depicts the sequence of acoustic observations $\mathbf{O}_{1:T} = [\mathbf{o}_1, \ldots, \mathbf{o}_T]$ extracted from the raw waveform from Figure 3.1 (a) (also see Section 3.5), which together with the underlying initial alignment of states $\mathbf{Q}_{1:T}$ (Figure 3.1 (b)) are used to estimate the parameters of the acoustic model $p(\mathbf{O}|\boldsymbol{w})$.

Similarly, as illustrated in Figure 3.1 (d), one also estimates a language model $P(\boldsymbol{w})$ to obtain a probability distribution over possible word sequences (see Section 3.4). At the recognition stage, contributions from the acoustic and language models are combined to search for the most likely sequence of HMM states that explain the observed acoustic sequence. This sequence can be then mapped to phonemes, words and whole sentences. The result of this stage is a recognition hypothesis – the most likely word sequence $\boldsymbol{w}^*$ in (3.1). This search process is briefly described in Section 3.6.

[2]This is not an unrealistic assumption and in practice, before expanding models to more suitable representation of acoustic space, one usually builds such an initial system.

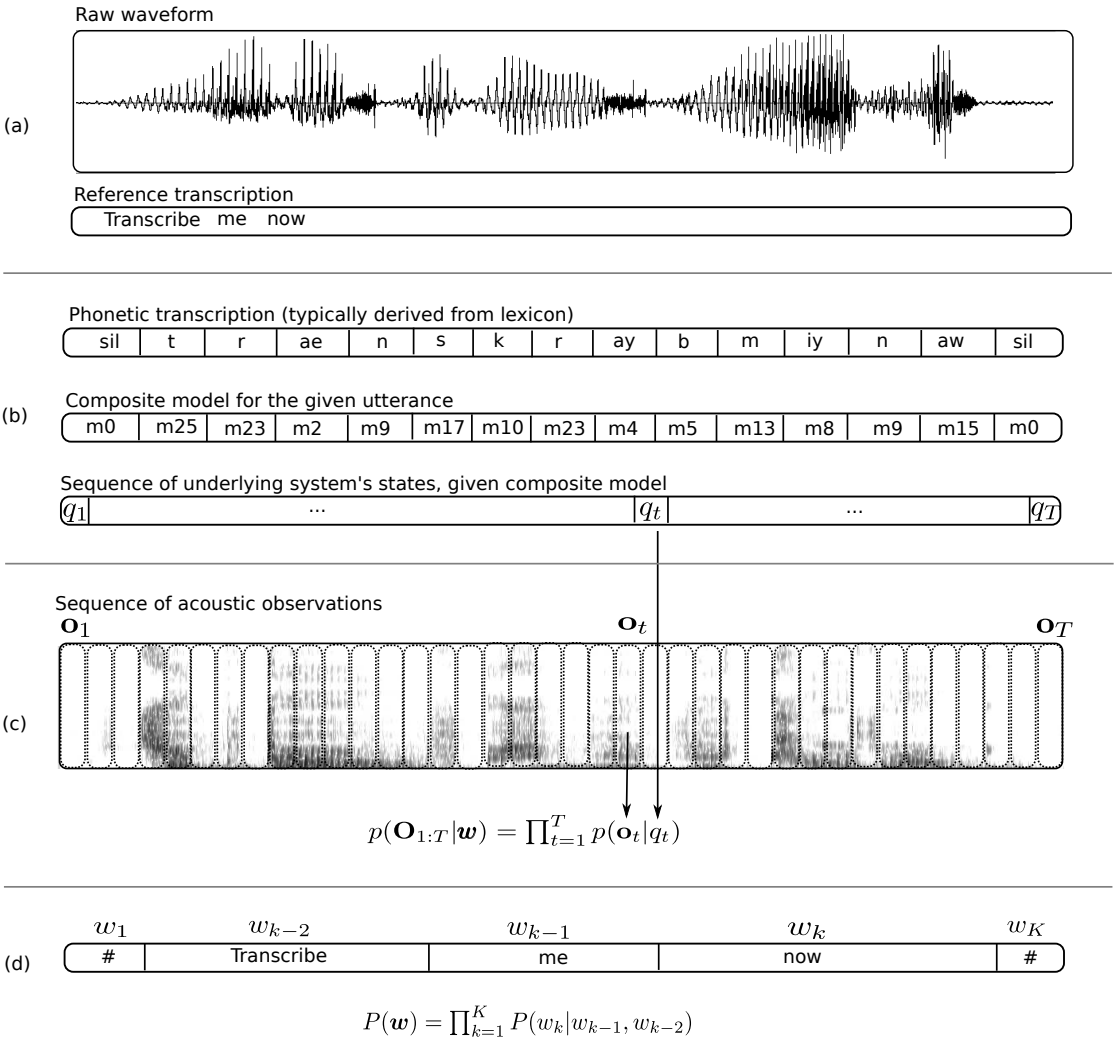


Figure 3.1: An analysis of a recognition of an example phrase "Transcribe me now". See text for description. Image adapted based on an idea in Young [2013].

3.3 Hidden Markov acoustic models

In this work, we are concerned with hidden Markov model (HMM) based acoustic models, in which the underlying process of the system's states is *hidden*. Refer to Figure 3.1 and note that the exact sequence of states can be inverted to recover desired higher level structures (sequences of phonemes and words).

Formally, an N-state HMM, as depicted in Figure 3.2, is parametrised by two sets of parameters:

- **State transition probabilities** - denoted by matrix $\boldsymbol{A}$ specify the prob-

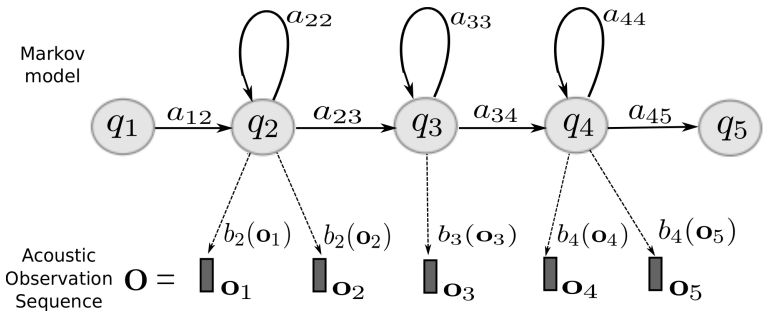


Figure 3.2: A left-to-right hidden Markov model comprising of 5 states. The first and the last state are non-emitting and are used to facilitate construction of composite HMMs. Transition probabilities implement an exponentially decreasing duration model, i.e. probability of remaining in state j for exactly t consecutive time steps is $d_j(t) = a_{jj}^{t-1}(1 - a_{jj})$.

abilities of moving from one state to another in an HMM, as follows:

$$(A)_{ij} = a_{ij} = P(q_t = j | q_{t-1} = i) \tag{3.4}$$

$$\text{subject to:} \sum_{j=1}^{N} a_{ij} = 1, \ \forall i = 1, \ldots, N. \tag{3.5}$$

- **State output probability distributions** - denoted by matrix $\boldsymbol{B}$ specify the likelihood of state j generating observation $\mathbf{o}_t$ at time t.

$$(\boldsymbol{B})_{jt} = b_j(\mathbf{o}_t) = p(\mathbf{o}_t | q_t = j) \tag{3.6}$$

  One can pick between different parametrisations of $b_j()$ yielding either a discrete HMM or a continuous density HMM when some form of probability density function is used. Continuous probabilistic mappings are nowadays the most common choices and we briefly describe them in Section 3.3.3.

An HMM (or a composite HMM when more than one model is involved) is thus parametrised by $\mathcal{M} = \{\{a_{ij}\}, \{b_j(\cdot)\}\}$ where the parameters of $\{b_j(\cdot)\}$ depend on the specific model used to estimate the output state probability distributions (see Section 3.3.3).

3.3.1 HMM assumptions

For computational reasons, the HMM topology used in ASR is typically restricted to be a first order Markov process; that is its state q_t at time t depends only on

the preceding state q_{t-1}:

$$P(q_t|q_{t-1}, q_{t-2}, \ldots, q_1) = P(q_t|q_{t-1}) \tag{3.7}$$

and observation vectors $\mathbf{o}_t$ at any particular time t are assumed to be *conditionally independent* of previous observations and states given the state q_t, as follows:

$$p(\mathbf{o}_t|\mathbf{o}_{t-1}, \ldots, \mathbf{o}_1, q_t, \ldots, q_1) = p(\mathbf{o}_t|q_t) \tag{3.8}$$

There has been some work investigating less restrictive families of graphical models [Bilmes and Bartels, 2005] or at least aimed at understanding how much each of the above assumptions affects acoustic modelling capabilities from the perspectives of speech recognition [Gillick et al., 2011] and synthesis [Henter et al., 2014]. However, for ASR there is not currently a competitive alternative offering more accurate and manageable large-scale systems. Notice that the incorrectness of the conditional independence assumption allows appended time-derivative features to increase the mutual information between neighbouring HMM state probability distributions [Bilmes, 2003, p. 37].

The above assumptions are important as they allow us to simplify the expression for calculating the acoustic likelihood over sequences of observations $\mathbf{O}$ and states $\mathbf{Q}$, factorising the total likelihood $p(\mathbf{O}|\boldsymbol{w}; \mathcal{M})$ under the model $\mathcal{M}$ as:

$$p(\mathbf{O}|\boldsymbol{w}; \mathcal{M}) = \sum_{\mathbf{Q} \in \boldsymbol{w}} \prod_{t=1}^{T} p(\mathbf{o}_t|q_t; \mathcal{M}) P(q_t|q_{t-1}; \mathcal{M}) \tag{3.9}$$

$$= \sum_{\mathbf{Q} \in \boldsymbol{w}} a_{q_0, q_1} \prod_{t=1}^{T} b_{q_t}(\mathbf{o}_t) a_{q_t, q_{t+1}} \tag{3.10}$$

Naive evaluation of the likelihood resulting from (3.10) is still prohibitively expensive requiring $\mathcal{O}(N^T)$ summations and multiplications. However, in practice one can rely on the Baum-Welch *forward* or backward [Baum and Eagon, 1967] recurrences which reduce this complexity to $\mathcal{O}(NT^2)$ steps. A simplified variant that only tracks the most likely state sequence ($\max_{\mathbf{Q} \in \boldsymbol{w}}$ instead of $\sum_{\mathbf{Q} \in \boldsymbol{w}}$ in (3.10)) is known as the *Viterbi* algorithm [Viterbi, 1967, Forney, 1973].

3.3.2 On the acoustic units and HMM states

Acoustic realisations of a particular phoneme can be very different depending on the acoustic context in which the phoneme is uttered. This effect is known as *co-articulation* (generic case) or *elision* (when the central phoneme is entirely omitted). Those observations motivated the building of HMMs for context-dependent

acoustic units that explicitly take into account those characteristics. The most common approach relies on building HMMs for *triphone* units, i.e. the units that model the central phoneme in the context of its immediate left and right neighbours. This typically leads to a dramatic explosion in the number of parameters that need to be estimated – for a typical language comprising of about 40 phones one would have to construct HMM models for 40^3 triphones, many of which would have few or no associated observations in the training data. The most successful method addressing this issue relies on clustering and tying the parameters of HMM states that describe similar acoustic information, performed in either a top-down [Young and Woodland, 1994] or a bottom-up [Hwang and Huang, 1993] manner.

3.3.3 Modelling state probability distributions in HMMs

In this section we review some of the most common approaches for modelling HMM state probability distributions.

3.3.3.1 GMM-HMM

A common choice for a probability density function (PDF) is the Gaussian mixture model (GMM). In the GMM-HMM approach the jth HMM state is modelled as a linear combination of M_j multivariate Gaussians:

$$b_j(\boldsymbol{o}_t) = p(\boldsymbol{o}_t|j; \{c_{jm}, \mu_{jm}, \boldsymbol{\Sigma}_{jm}\}_{m=1}^{M_j}) = \sum_{m=1}^{M_j} c_{jm}\mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_{jm}, \boldsymbol{\Sigma}_{jm}) \tag{3.11}$$

subject to:

$$\sum_{m=1}^{M_j} c_{jm} = 1 \text{ and } c_{jm} \geq 0 \tag{3.12}$$

where the likelihood of an observation $\mathbf{o}_t$ being generated by the particular component is given as:

$$\mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) = \frac{1}{\sqrt{(2\pi)^D|\boldsymbol{\Sigma}_j|}}\exp\left\{-\frac{1}{2}(\boldsymbol{o}_t - \boldsymbol{\mu}_j)^T\boldsymbol{\Sigma}_j^{-1}(\boldsymbol{o}_t - \boldsymbol{\mu}_j))\right\} \tag{3.13}$$

Parameters $\boldsymbol{\mu}_m$ and $\boldsymbol{\Sigma}_m$ denote the mean and the covariance matrix of the mth component, $\boldsymbol{o}_t$ it the observation vector at time t and D is the dimensionality of acoustic vector $\mathbf{o}_t$. The problem of estimation of GMM-HMM parameters $\boldsymbol{\theta} = \{\{a_{ij}\}, \{\boldsymbol{c}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}\}$ from data will be collectively addressed in Section 3.3.4.

3.3.3.2 Subspace Gaussian Mixture Models

In a conventional GMM-based acoustic model, parametrisations of all mixtures in the system are state-dependent. This poses a challenge when only a small amount of data is assigned to a given state. State-tying mechanisms, mentioned in Section 3.3.2, provide a partial remedy by merging acoustically similar states. However, this is usually not sufficient and it is a common practice to ignore dependencies within an acoustic vector allowing the use of diagonal covariance GMMs. The subspace Gaussian Mixture Model (SGMM) [Povey et al., 2011a] relaxes the structure of how the GMMs are organised in the system by managing a pool of fewer (typically several hundred) full-covariance Gaussian components. Those components are state-independent, which allows for more accurate estimation from more, possibly multi-lingual, data [Burget et al., 2010, Lu et al., 2014], which are then used to derive state-dependent PDFs.

3.3.3.3 ANN-HMM

The use of ANNs for modelling state output probabilities in HMM was proposed as an alternative to GMMs by [Bourlard and Wellekens, 1990]. In this hybrid ANN-HMM approach, a neural network is used to compute the posterior probability over HMM states given a window of acoustic observations $\bar{\mathbf{O}}_t = [\mathbf{o}_{t-c}^\top, \ldots, \mathbf{o}_t^\top, \ldots, \mathbf{o}_{t+c}^\top]^\top$. Those posteriors are further scaled by HMM state priors to obtain scaled-likelihoods [Bourlard and Morgan, 1994]. For the jth HMM state we have

$$b_j(\mathbf{o}_t) = p(\mathbf{o}_t|q_t = j) = \frac{P_{ANN}(j|\bar{\mathbf{O}}_t)P(\mathbf{o}_t)}{P(j)} \propto \frac{P_{ANN}(j|\bar{\mathbf{O}}_t)}{P(j)} \tag{3.14}$$

where $P_{ANN}(q_t|\bar{\mathbf{O}}_t)$ is estimated by a some form of connectionist model $f(\bar{\mathbf{O}}_t;\boldsymbol{\theta})$, acting on $\bar{\mathbf{O}}_t$ and parameterised by $\boldsymbol{\theta}$. Many parametric forms of $f(\bar{\mathbf{O}}_t;\boldsymbol{\theta})$ have been studied for ASR, including a variety of feed-forward and recurrent models with different transfer functions and connectivity patterns. $P(j)$ is calculated from training state-level forced alignments and smoothed (in this work) with Laplace discounting for numerical stability:

$$P(j) = \frac{\mathcal{C}(j) + \alpha}{\sum_{j'} \mathcal{C}(j') + \alpha|\mathcal{Q}|} \tag{3.15}$$

$\mathcal{C}(\cdot)$ denotes the number of occurrences of a state j in the training alignments, α (set to 1 in this work) is a smoothing factor and $|\mathcal{Q}|$ is the number of leaves of a decision clustering tree (see Section 3.3.2).

Initially ANN-HMMs were built for context-independent (CI) acoustic models but attempts at handling context dependency have also been made by training an ensemble of distinct ANNs to estimate a set of conditional probabilities necessary to derive a context-dependent (CD) likelihood score [Morgan and Bourlard, 1992, Bourlard et al., 1992, Franco et al., 1994]. Example factorisations involved predicting a context-independent state (or phoneme) q_t given an observation $\bar{\mathbf{O}}_t$, and then a context dependent cluster unit c given previously predicted CI state q_t and observation $\bar{\mathbf{O}}_t$ (3.16), or the other way round using equation (3.17):

$$P(q_t, c_t|\bar{\mathbf{O}}_t) = P_{ANN}(q_t|\bar{\mathbf{O}}_t)P_{ANN}(c_t|q_t, \bar{\mathbf{O}}_t) \tag{3.16}$$

$$= P_{ANN}(c_t|\bar{\mathbf{O}}_t)P_{ANN}(q_t|c_t, \bar{\mathbf{O}}_t) \tag{3.17}$$

Recently [Dahl et al., 2012] proposed to use ANNs to directly model the posterior distribution over context-dependent HMM clustered tied-states inherited from a corresponding GMM-HMM system. The number of such states for a typical ASR scenario, depending on the amount of training data, ranges from several hundred up to thousands or tens of thousands of acoustic classes. However, as has been found in [Dahl et al., 2012, Seide et al., 2011] this distribution can be reliably approximated with deeper and wider networks and more data. Initially, this was also attributed to restricted Boltzmann machine based layer-wise pre-training by [Hinton et al., 2006], however, if the amount of data is sufficient (say 50 or more hours), pre-training has a rather negligible impact on final accuracy [Seide et al., 2011] compared to other aspects like depth and/or CI vs. CD states. An ANN-HMM system is illustrated in Figure 3.3. In a sense, much of this progress was made possible due to significant advances in hardware architecture, especially the usage of general purpose graphic processing units (GP-GPU). This allowed the speeding up matrix of algebra computations, which enabled training of large ANN models in a reasonable amount of time (compared to highly parallelisable GMM-HMM systems). A systematic review of some of the many recent developments within hybrid ANN-HMM framework [Bourlard and Morgan, 1994] can be found in book positions of Yu and Deng [2014] and Li et al. [2015].

A considerable advantage of the ANN-HMM approach is its more efficient parameterisation. Contrary to the GMM-HMM approach, where distinct sets of parameters are responsible for modelling different acoustic events, acoustic models based on ANNs share parameters used across all competing classes. This encourages greater parameter reuse allowing poorly represented data classes (HMM

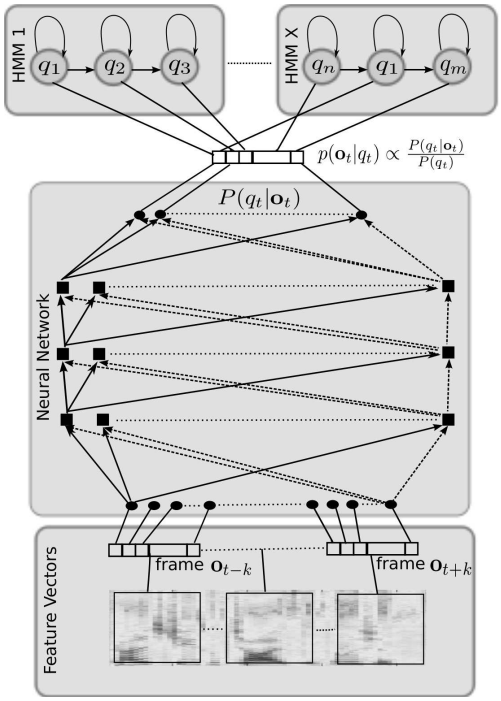


Figure 3.3: Architecture of an hybrid acoustic model using ANN to estimate the posterior probability over HMM-states. HMM-states can be either context-dependent or context-independent.

states) to make use of a shared discriminatory knowledge discovered during learning from data assigned to competing classes. As a result, larger context-dependent trees can be used to better cover the acoustic space without issues arising from data sparsity, as it is the case when the representation is not shared [Li et al., 2012, Liao et al., 2013].

3.3.3.4 KL-HMM

The Kullback-Leibler HMM (KL-HMM) [Aradilla et al., 2008] relies on modelling HMM state emission probabilities using a phone posterior distribution estimated by an ANN. This is a generalisation of hybrid systems, enabling an ANN model distribution across acoustic units that can be independent of a particular HMM state (unlike in the hybrid ANN-HMM framework where the outputs of the ANN are tied to particular states). This is done by looking at the KL divergence between posterior predictions obtained using an ANN and the expected distribution of HMM states, which can be learned. Most of the experiments along this avenue were performed for context–dependent or context–independent phonemes. For those experiments the KL-HMM systems offered better results when compared

to context-independent hybrid systems. This effect was particularly pronounced when building extremely low-resource systems (5 minutes of transcribed data or so) in which an ANN could be estimated from multi-lingual data [Imseng et al., 2012] working directly as an acoustic model within KL-HMM.

3.3.4 Estimating parameters of HMM-based acoustic model

3.3.4.1 GMM-HMM

Here we discuss training of HMM models in which state output probabilities are assumed to be estimated by either GMM or ANN models. For the GMM-HMM scenario the basic criterion for training parameters is maximum-likelihood (ML):

$$\mathcal{F}_{ML}(\boldsymbol{\theta}_u) = \frac{1}{U}\sum_{u=1}^{U} \log p\left(\mathbf{O}_u|\boldsymbol{w}_u^{ref};\boldsymbol{\theta}_u\right) \tag{3.18}$$

where $\boldsymbol{\theta}_u = \{\{a_{ij}\}_u, \{\boldsymbol{c},\boldsymbol{\mu},\boldsymbol{\Sigma}\}_u\}$ denotes the parameters of a sequence of concatenated GMM-HMM models (or a composite HMM model) built for the reference transcript $\boldsymbol{w}_u^{ref}$ and $\mathbf{O}_u$ acoustic sequence. The optimisation is then typically carried out using the forward-backward Baum-Welch recursions [Baum and Eagon, 1967] which is an instance of an expectation-maximisation (EM) algorithm [Dempster et al., 1977] that is guaranteed to not decrease the likelihood between successive iterations. Equation (3.18) maximises the likelihood of a subset of parameters associated with phonetic models of the utterance u and does not take into account rival models which may also generate those observations well; as a result ML may result in poor discriminative properties. It is a common practice to refine generative acoustic model parameters in a sequence discriminative manner. Many techniques have been developed for this purpose aimed at either maximising the mutual information (MMI) between the competing models [Bahl et al., 1986, Povey, 2003] or minimising the expected error rates [Juang and Katagiri, 1992, Kaiser et al., 2000, Povey, 2003]. In this work we either use a boosted variant of MMI [Povey et al., 2008] (GMM-HMM systems) or perform state-level minimum Bayes risk (MBR) training (ANN-HMM systems). The generic MBR criterion is defined as:

$$\mathcal{F}_{MBR}(\boldsymbol{\theta}) = \frac{1}{U}\sum_{u=1}^{U}\sum_{\boldsymbol{w}} P(\boldsymbol{w}|\mathbf{O}_u;\boldsymbol{\theta})A(\boldsymbol{w},\boldsymbol{w}_u^{ref}) \tag{3.19}$$

$$= \frac{1}{U}\sum_{u=1}^{U}\frac{\sum_{\boldsymbol{w}} p(\mathbf{O}_u|\boldsymbol{w};\boldsymbol{\theta})^{\kappa}P(\boldsymbol{w})A(\boldsymbol{w},\boldsymbol{w}_u^{ref})}{\sum_{\boldsymbol{w}'} p(\mathbf{O}_u|\boldsymbol{w}';\boldsymbol{\theta})^{\kappa}P(\boldsymbol{w}')} \tag{3.20}$$

where κ is an acoustic scale and $A(\boldsymbol{w}, \boldsymbol{w}_u^{ref})$ denotes the loss function of the word sequence $\boldsymbol{w}$ with respect to the reference sequence $\boldsymbol{w}_u^{ref}$ of the utterance u. The loss $A(\cdot, \cdot)$ may be computed at different levels (word, phone, state) leading to different objectives [Gales and Young, 2008]. GMM-HMM parameters for discriminative training are estimated using an extended Baum-Welch procedure [Gopinath, 1998].

3.3.4.2 ANN-HMM

In the case of a hybrid system, one typically estimates ANN parameters such that it maximises the posterior probability of predicting the correct tied state $q_t \in \mathcal{Q}$ at each time-step t. This is expressed as minimising the KL divergence between two distributions – the empirical one estimated from training data $\boldsymbol{t}$ and the one predicted by the ANN. This is also known as *cross-entropy* and is defined as:

$$\mathcal{F}_{CE}(\boldsymbol{\theta}) = -\frac{1}{U}\sum_{u=1}^{U}\frac{1}{T_u}\sum_{t=1}^{T_u}\log P_{ANN}(q_t|\bar{\boldsymbol{O}}_{ut};\boldsymbol{\theta}) \tag{3.21}$$

Supervision targets for each frame t in each utterance u, $\boldsymbol{t}_{ut}$, are obtained by force-aligning acoustic observations with the reference transcripts using a GMM-HMM system[3]. One also usually uses transition matrices from GMM-HMM system, though those have a negligible impact on accuracy and one can assume a uniform transition model [Bourlard and Morgan, 1994].

Criterion (3.21) typically operates at the frame level (see [Hennebert et al., 1997] for a sequential formulation using Baum-Welch forward-backward recursions), contrary to sequential formulation of (3.18), and the resulting model is discriminative at the frame-level in the MMI sense[4]. One can also train ANNs to discriminate at the sequence level, following the same objective as for GMM-HMM system. In this work we minimise the expected loss with MBR as defined in (3.20). The ANN parameters with either criteria can be estimated with any gradient based optimiser. In this work we use stochastic gradient descent [Bishop, 2006].

[3]Dependence on GMM-HMM is not required though. One can build context-dependent ANN-HMM systems starting from uniform context-independent alignments and context-independent ANN models which can be iteratively re-estimated and expanded to context-dependent models [Zhang and Woodland, 2014a, Senior et al., 2014]

[4]As one updates all ANN parameters taking into account all competing classes.

3.4 Language model

The language model (LM), $P(\boldsymbol{w})$ in (3.2), estimates a probability distribution over sequences of words $\boldsymbol{w} = [w_1, w_2, ..., w_K]$. The dominant technique for approximating such language grammars relies on n-gram models [Damerau, 1971], expressed as:

$$P(\boldsymbol{w}) = \prod_{k=1}^{K} P(w_k|w_{k-1}, w_{k-2}, ..., w_{k-n+1}) \tag{3.22}$$

where n denotes the n-gram order, or number of words used in conditioning in (3.22). The LMs are usually trained to maximise the likelihood [Dempster et al., 1977] of training data which for higher order n-gram usually leads to very sharp distributions with many zero probabilities for unobserved word sequences. For this reason robust language modelling requires additional smoothing techniques. These can be grouped, in a non-exclusive way, as follows:

- Back-off: family of techniques proposed by [Katz, 1987]. Contrary to Laplacian, discounting back-off assigns the probability mass unevenly to unseen word tokens in proportion to the probability of a lower-order n-gram, $P(w_k|w_{k-1}, w_{k-2}) \approx P(w_k|w_{k-1})$.

- Discounting: relies on assigning some of the probability distribution mass to n-gram tokens unseen at training stage. The one that is used across this work uses [Kneser and Ney, 1995] smoothing.

- Interpolation: relies on linearly weighting the same or different order n-gram language models. This approach differs from back-off methods in the way it handles n-gram with zero counts.

Class-based language models [Brown et al., 1992] (equation (3.23)) factorises the prediction of w_k on the class c_k the word was assigned to and the history of preceding classes. The number of classes in a typical scenario is up to several hundreds, resulting in less severe data-sparsity issues and offering a more convenient approach of vocabulary expansion. Estimation, similarly to n-gram models, can be carried out in maximum likelihood manner.

$$P(\boldsymbol{w}) = \prod_{k=1}^{K} P(w_k|c_k)P(c_k|c_{k-1}, ..., c_{k-n+1}) \tag{3.23}$$

In recent years neural networks have become a popular choice in modelling language grammars. The idea, initially proposed in the context of feed-forward

ANNs by Bengio et al. [2003], relies on mapping a discrete n-gram word distribution into a continuous space. Recently this idea has been extended to recurrent ANNs by Mikolov et al. [2010]. From an ASR perspective, ANN-based LMs are primarily used in a post-processing step to re-score ASR hypotheses. There has been some work to address the problem of efficient handling of large vocabularies by RNNs (say 50k+) in which case one may use class-based factorisation, similar to the one in equation (3.23) and the factorisation proposed for context-dependent acoustic modelling for ANN by Bourlard et al. [1992] (see also Section 3.3.3.3).

For some systems we will report perplexity (PPL) of the corresponding LMs. PPL is typically estimated on some held out task-related data, and lower perplexity means the LM will generalise better when applied to predict unseen data. This usually results in better ASR performance. PPL is defined as:

$$\text{PPL} = \exp\left\{-\frac{1}{K}\sum_{k=1}^{K}\log\left(P(w_k|w_{k-1}, w_{k-2}, ..., w_{k-n+1})\right)\right\} \tag{3.24}$$

3.5 Feature extraction

Feature extraction from the raw waveform signal, as designed for ASR, seeks compact representations that minimize variability across speakers and environmental acoustic conditions. At the same time, it aims to retain good discriminative information between words. Typically one also seeks a representation of the acoustic signal that fits the target classifier capabilities, for example, features that are about to be modelled by diagonal covariance GMM-HMM models should be Gaussian and uncorrelated.

The most common approach relies on treating a speech signal as a piecewise stationary process in which chunks of the waveform (typically 25ms long) are processed in 10ms shifts. Typically each such window is transformed into the spectral domain using short-time fast Fourier transform (FFT) followed by transformation to power-spectra and smoothed by 20-40 *mel filter-bank* filters, in order to perform an auditory-based warping of the frequency axis to account for the frequency sensitivity of human hearing system. Those smoothed power-spectra are further logarithmically compressed and are referred to as mel-filter bank (FBANK) features, and are sometimes used in this thesis directly to train ANN acoustic models. FBANK features are further processed for diagonal GMMs with a decorrelating DCT transform, resulting in mel-frequency cepstral coeffi-

cients features (MFCC) [Mermelstein, 1976]. Another popular acoustic feature extractor, perceptual linear coefficients (PLP) [Hermansky, 1990], relies on using *bark* scale to compute the filter-bank filters (instead of mel scale) followed by a linear predictive analysis, from which one then derives a cepstral representation. It is a common practice to augments *static* features (described above) with *dynamic* features spanning longer time windows using either difference method $\Delta \boldsymbol{o}_t^r = \boldsymbol{o}_{t+1}^r - \boldsymbol{o}_{t-1}^r$ or linear regression:

$$\Delta \boldsymbol{o}_t^r = \frac{\sum_{\delta=1}^{\Delta} \delta(\boldsymbol{o}_{t+\delta}^r - \boldsymbol{o}_{t-\delta}^r)}{2\sum_{\delta=1}^{\Delta} \delta^2} \tag{3.25}$$

One can derive dynamic features of an arbitrary order, however in this thesis we only use the first and second–order time derivatives, which are often referred to as *delta* and *acceleration* coefficients, respectively. The target acoustic vector at a time t – $\mathbf{o}_t$ – is build by concatenating static coefficients with time-derivatives:

$$\mathbf{o}_t = \begin{bmatrix} \boldsymbol{o}_t^r \\ \Delta \boldsymbol{o}_t^r \\ \Delta^2 \boldsymbol{o}_t^r \end{bmatrix} \tag{3.26}$$

3.5.1 ANNs as feature extractors

ANNs are often used in ASR as an explicit extractor of discriminative features on top of which other classifiers, not necessarily ANN, are built. Such a combination of ANN with GMM-HMM systems is known as the *tandem* approach [Hermansky et al., 2000]. The original formulation used posteriogram features, i.e. the posterior distribution across mono-phones estimated by ANN and appropriately post-processed (decorrelated, dimensionality-reduced, made more Gaussian) to better fit GMM-HMM models. There have been many improvements proposed since, including bottleneck features [Grezl et al., 2007] extracted from a low-dimensional hidden layer and various knowledge-motivated ANN ensembles, see [Morgan, 2012] for review. Those features can be used in a standalone manner or to augment the standard acoustic features. The potential advantage behind *tandem* systems is that it allows the use of much of the machinery originally developed for classical GMM-HMM systems, including adaptation, sequence discriminative training and noise compensation. A tandem system architecture, in which ANN-based features are concatenated with standard acoustic features, is illustrated in Figure 3.4.

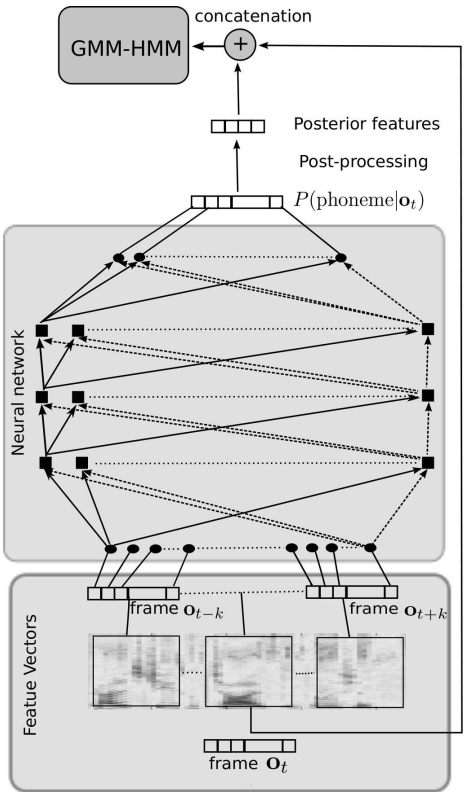


Figure 3.4: Architecture of contemporary tandem acoustic models

3.6 Decoding

Once acoustic and language models have been estimated, one can proceed and perform recognition, as in equation (3.3). This involves obtaining and combining their respective scores. Because the acoustic model severely over-estimates the confidence of a sequence of acoustic events, one must scale up the contributions of the language model (or down-scale the acoustic model). This is done by a hyper-parameter κ. λ is a word insertion penalty. Both are optimised on development data. In experiments reported in this thesis, $\kappa \in [10, 15]$ and $\lambda = 0$. The recognition process is summarised in equations (3.27)-(3.29), where (3.29) represents the Viterbi approximation by taking into account only the most likely state sequence.

$$\boldsymbol{w}^* = \arg\max_{\boldsymbol{w}} \log P(\boldsymbol{w}|\mathbf{O}) \tag{3.27}$$

$$= \arg\max_{\boldsymbol{w}} \{\log p(\mathbf{O}|\boldsymbol{w}; \mathcal{M}) + \kappa \log P(\boldsymbol{w}) + \lambda\} \tag{3.28}$$

$$\approx \arg\max_{\boldsymbol{w}} \left\{\log \left(\max_{\mathbf{Q}} p(\mathbf{O}, \mathbf{Q}|\boldsymbol{w}; \mathcal{M})\right) + \kappa \log P(\boldsymbol{w}) + \lambda\right\} \tag{3.29}$$

3.7 Evaluation

Once the system has been trained, it is important to say how well it performs under certain testing conditions, often in relation to other systems. To do so, one usually expresses the performance of the ASR system in terms of an accuracy metric which allows a direct comparison between systems.

The most common measure to score ASR systems relies on calculating word error rates (WER) expressed as the percentage ratio of the number of incorrectly recognised words with respect to the reference word-level transcript. One usually distinguishes between three types of errors referred to as *substitutions* (S), *deletions* (D) and *insertions* (I). As such, given the total number of words to be recognised in the reference transcript is N, the WER is expressed as in equation (3.30):

$$\text{WER} = \frac{\text{S} + \text{I} + \text{D}}{\text{N}} \times 100\% \tag{3.30}$$

WER is a convenient metric allowing to compare qualities of different ASR systems on some fixed held-out test data.

3.8 Summary

We have reviewed HMM based acoustic models with a particular interest in the integration of ANNs into the framework. This can be done at two levels, where the ANN is used to either extract some form of discriminative features from acoustic observations which usually leads to systems we refer to as *tandem* systems, where a GMM acoustic model is trained on top. The other approaches handle HMM state emission probabilities directly, this leads to systems such as KL-HMM and hybrid. The remainder of this thesis is primarily concerned with hybrid ASR systems.

Chapter 4

Corpora and Baseline Systems

This chapter collates descriptions, systems details and baseline numbers for the corpora used through this thesis.

4.1 Introduction

Our investigations in this dissertation are focused on three avenues related to acoustic modelling: adaptation; learning from multiple acoustic channels; and estimation of acoustic models from low-resource data. Each part places different constraints with respect to the available data resources and the working conditions of the resulting systems. This also determines the use of particular corpora for certain types of experiments. Where possible we try to make our findings stronger by evaluating how well our systems generalise to different benchmark corpora.

In this chapter we outline the corpora and baseline experimental systems used through this work. The TED Lectures corpus, used in all adaptation experiments, is described in section 4.2. Distant speech recognition experiments make use of AMI and ICSI meetings data described in sections 4.3.1 and 4.3.2, respectively. Switchboard conversational telephone speech data is outlined in section 4.4. We also performed some experiments on factorisation of environmental acoustic conditions using Aurora-4, outlined in section 4.5. The low-resource experiments make use of the GlobalPhone multilingual corpus which is described in section 4.6. We add a note on the statistical significance of the experimental results in section 4.7.

Table 4.1: TED Lecture statistics and baseline %WER. Notice GMM-HMM models are adapted to speakers while ANN-HMM are not. Reported p-values from significance tests were performed for GMM-HMM vs. ANN-HMM pairs and `4gm-751MW` LM.

Set	Time (hours)	Num. of Speakers	GMM-HMM %WER		ANN-HMM %WER		p_v
			`3gm-p07`	`4gm-751MW`	`3gm-p07`	`4gm-751MW`	
Training	143	788	-	-	-	-	
`dev2010`	1.5	8	20.0	17.0	18.3	15.4	<0.001
`tst2010`	2.5	11	19.0	16.3	18.0	15.0	<0.001
`tst2011`	1.1	8	15.1	13.1	14.6	12.1	<0.001
`tst2013`	3.9	28	30.7	26.3	25.9	22.1	<0.001

4.2 TED Lectures

The TED lectures corpus comprises of publicly available TED talks[1] [Cettolo et al., 2012]. Each talk is in most cases dominated by a single speaker and is between 5 and 15 minutes long. In this work the data was prepared following the ASR evaluation protocol of International Workshop on Spoken Language Translation (IWSLT)[2], in particular, we work with the testing conditions defined for IWSLT2012 [Federico et al., 2012] and IWSLT2013 [Cettolo et al., 2013] evaluation campaigns. The training data, unless explicitly stated otherwise, consists of speech from 813 talks recorded before the end of 2010. An initial pre-processing involved lightly supervised alignment of acoustic data with available on-line transcripts [Stan et al., 2012]. This resulted in about 143 hours of in-domain training material. We present results on four predefined IWSLT test sets: `dev2010`, `tst2010`, `tst2011` and `tst2013`. The last one, `tst2013`, was provided without manual segmentations and automatic segmentations were obtained following [Sinclair et al., 2014]. Corpus statistics and baseline WERs are reported in Table 4.1. Notice, in this chapter we mostly report results for the baseline unadapted ANN-HMM systems, and their adapted variants are reported later in Chapters 5 and 6.

4.2.0.1 Acoustic models

Acoustic models were trained on perceptual linear prediction (PLP) [Hermansky, 1990] acoustic features with first– and second–order time derivatives (39 coeffi-

[1] `www.ted.com`
[2] `www.iwslt.org`

Table 4.2: Language model perplexities on development set

Language model	PPL
`3gm-2.4MW` (TED)	183.2
`3gm-312MW`/ `3gm-751MW` (TED+OOD)	125.1 / 124.9
`4gm-2.4MW` (TED)	179.9
`4gm-312MW`/ `4gm-751MW` (TED+OOD)	114.9 / 113.4

cients in total), which were globally normalised to have zero mean and unit variance. GMM-HMM models were trained in a speaker adaptive manner (SAT) using constrained maximum likelihood linear regression (CMLLR) transforms [Gales, 1998]. The parameters of this intermediate SAT-GMM-HMM system were further refined in sequence discriminative fashion using boosted maximum mutual information (BMMI) criterion [Povey et al., 2008]. From now on, we shall refer to this final system utilising 12,000 tied-states and 192,000 Gaussian components as GMM-HMM.

The ANN models had 6 hidden layers with 2048 units in each layer. The input context window was comprised of 9 consecutive PLP acoustic vectors. The ANN was used in a hybrid setup to estimate the distribution over tied-states obtained from GMM-HMM system, which was also used to produce targets for ANN training. ANNs were randomly initialised and pre-training was not performed. ANN parameters are optimised with stochastic gradient descent in 256 element mini-batches to minimise the negative log-posterior probability of predicting a correct tied-state at each time-frame. Unless stated otherwise, most of the ANN models in this work were trained with the `newbob` performance scheduler [Renals et al., 1992, Senior et al., 2013]. That is, the model is learnt at a fixed rate (0.08 in this work) as long as the frame accuracy between two successive epochs on a development set keeps improving (here by at least 0.25%). Otherwise the learning rate is halved each epoch and learning stops if improvement falls below 0.1%.

4.2.1 Language models

We use two types of language models (LM) for TED through this thesis, each LM utilises out-of-domain (OOD) texts giving in total 312 or 751 million training words (MW). The perplexities of those LMs on `dev2010` are outlined in Table 4.2. We use a pruned version of the `3gm-312MW` LM to obtain initial recognition hy-

potheses (denoted by `3gm-312MW.p07`, where a pruning threshold is set to 10^{-7}). Lattices are then re-scored using an unpruned 3 or 4-gram LM. The vocabulary was limited to 64,000 words.

4.3 AMI and ICSI meetings corpora

We describe those corpora together as the underlying ASR systems share many common characteristics, and ICSI is used as an auxiliary dataset for complementary experiments, where necessary.

4.3.1 AMI

The AMI[3] (Augmented Multiparty Interaction) corpus [Carletta, 2007, Renals et al., 2007] contains around 100 hours of meetings recorded in specifically equipped instrumented meeting rooms at three sites in Europe (Edinburgh, IDIAP, TNO). There are two types of meetings—scenario based, where four speakers act out certain predetermined roles of a design team (project manager, designer, etc.), as well as non-scenario-based which are natural spontaneous meetings on a range of topics. The scenario-based meetings make up about 70% of the corpus. Each meeting usually has four participants and the meetings are in English, albeit with a large proportion of non-native speakers. The acoustic signal was captured by multiple microphones including individual head microphones (IHM), lapel microphones, and one or more microphone arrays. Each recording site use a primary 8-microphone uniform circular array of 10 cm radius, as well as a secondary array whose geometry varied between sites. In this work we use the primary array and refer to it as the multiple distant microphones (MDM) variant. Experiments with a single distant microphone (SDM) make use of the first microphone of the primary array.

Most previous ASR research using the AMI corpus [Hain et al., 2012, Grezl et al., 2007] has been in the context of the NIST Rich Transcription (RT) evaluations, where the AMI data was used together with other meeting corpora. In order to perform more controlled experiments with identical microphone array configurations, we have defined a 3-way partition of the AMI corpus into train, development, and test sets. This partition makes about 78 hours of speech avail-

[3] `http://corpus.amiproject.org`

able for training, and holds out about 9 hours each for development and test sets. All the three sets contain a mix of scenario- and non-scenario-based meetings, and are designed such that no speaker appears in more than one set. The definitions of these sets have also been made available on the AMI corpus website and are used in the associated Kaldi recipe[4]. We use the segmentation provided with the AMI corpus annotations (version 1.6.1). In this work, we consider all segments (including those with overlapping speech), and the speech recognition outputs are scored by the `asclite` tool [Fiscus et al., 2006] following the NIST RT[5] recommendations for scoring simultaneous speech.

4.3.2 ICSI

The ICSI[6] meeting corpus [Janin et al., 2003] comprises around 72 hours of speech recorded between 2000 and 2002 at the International Computer Science Institute during weekly speech research group meetings. The general setup is similar to that of AMI and comes with speech acoustics captured in close-talk and distance using both low and high quality microphones. Contrary to the circular microphone arrays used in AMI, distant speech in ICSI was captured using four independent microphones, placed to roughly approximate a linear tabletop array.

The original ICSI data does not provide pre-defined training and testing partitions. Instead, it was mainly used as a training resource for RT evaluations where some held-out parts were transcribed and used in connection with other meeting data to form test sets of RT evaluation campaigns. Those partitions, however, are not readily available and in this work we take 5 complete meetings out of training data for development and testing purposes, we will refer to them as `icsidev` and `icsieval` hereafter[7]. For simplicity of exposition, unless stated otherwise, we report results using all segments, including those with overlapping speakers and follow the same scoring procedure as in AMI (see Section 4.3.1).

4.3.3 Acoustic models

For the IHM configuration (AMI only), 7 frames (3 on each side of the current frame) of 13-dimensional MFCCs (C0-C12) are spliced together and projected

[4] `https://github.com/kaldi-asr/kaldi/tree/master/egs/ami/`
[5] `http://nist.gov/speech/tests/rt/2009`
[6] `http://catalog.ldc.upenn.edu/LDC2004S02`
[7] Meeting IDs: `dev` {Bmr021 and Bns001}, `eval` {Bmr013, Bmr018 and Bro021}

Table 4.3: Word error rates (%) for the GMM and DNN acoustic models for various microphone configurations.

System	Microphone configurations				
	IHM	MDM8	MDM4	MDM2	SDM
Development set (`amidev`)					
GMM BMMI on LDA/STC	30.2 (SAT)	54.8	56.5	58.0	62.3
ANN on LDA/STC	26.8 (SAT)	49.5	50.3	51.6	54.0
Evaluation set (`amieval`)					
GMM BMMI on LDA/STC	31.7 (SAT)	59.4	61.2	62.9	67.2
ANN on LDA/STC	28.1 (SAT)	52.4	52.6	52.8	59.0

down to 40 dimensions using linear discriminant analysis (LDA) [Haeb-Umbach and Ney, 1992] and decorrelated using a single semi-tied covariance (STC) transform [Gales, 1999], sometimes also termed a maximum likelihood linear transform (MLLT). These features are referred to as LDA/STC. Both the GMM-HMM and ANN-HMM acoustic models are speaker adaptively trained (SAT) on these LDA/STC features using a single CMLLR transform estimated per speaker. The GMM-HMM systems provide the state alignments for training the ANNs. Additionally, in some experiments the ANNs are trained on 40-dimensional log mel filterbank (FBANK) features appended with delta and acceleration coefficients. The state alignments used for training the ANNs on FBANK features are the same as those used for the LDA/STC features.

For the MDM (AMI and ICSI) experiments, we use delay-sum beamforming on either 2, 4, or 8 uniformly-spaced array channels (AMI) or 4 tabletop microphones (ICSI) using the BeamformIt toolkit [Anguera et al., 2007]. In both the SDM and MDM case, the audio is then processed in a similar fashion to the IHM configuration. The major difference between the IHM and SDM/MDM configurations is that when audio is captured with distant microphones, it is not realistically possible to ascribe a speech segment to a particular speaker without using speaker diarisation. As such, the SDM/MDM experiments, if not stated otherwise, do not use any form of speaker adaptation or adaptive training.

The GMM-HMM systems are trained on the speaker adapted LDA/STC features for the IHM case, or on the unadapted features for the SDM/MDM case, using the BMMI criterion. The number of tied-states is roughly 4000 in all con-

figurations, and each of the GMM-HMM systems has a total of 80,000 Gaussians. These are then used to provide the state alignments for training the corresponding ANNs using either the LDA/STC features or the FBANK features. The baseline results are summarised in Table 4.3.

4.3.4 Lexicon and language model

The experiments reported in this thesis used the 50,000 word AMI pronunciation dictionary as in Hain et al. [2012][8]. An in-domain trigram language model (LM) is estimated from the 801K words of the training AMI transcripts, which is then interpolated with an other trigram LM estimated from 22M words of the Fisher English transcripts. The LMs are estimated using interpolated Kneser-Ney smoothing. The in-domain AMI LM has an interpolation weight of 0.78, the Fisher LM gets a weight of 0.22. The final interpolated LM achieves a perplexity of 80 on the development set. This AMI LM forms our background LM for ICSI meeting, and is additionally interpolated with ICSI in-domain LM.

4.4 Switchboard

Switchboard corpus of conversational telephone speech[9] [Godfrey et al., 1992] is a popular benchmark for ASR purposes. We start with the Kaldi GMM recipe[10] [Vesely et al., 2013a, Povey et al., 2011b], using Switchboard–1 Release 2 (LDC97S62). Our baseline unadapted acoustic models were trained on either MFCC and/or LDA/STC features. The results are reported on the full Hub5 âĂŹ00 set (LDC2002S09) which we will refer to as `eval2000`. The `eval2000` contains two types of data, Switchboard (SWBD) – which is better matched to the training data – and CallHome English (CHE). Our baseline results, reported in Table 4.4, use 3-gram LMs estimated from Switchboard and Fisher data. The dictionary for this task has 30 000 words.

[8]The dictionary from Hain et al. [2012] initially used in the experiments was a proprietary component and the Kaldi recipe is based on an open-source CMU dictionary. This change has only a negligible impact on final accuracies

[9]`ldc.upenn.edu`

[10]To stay compatible with our published adaptation work on Switchboard [Swietojanski and Renals, 2016, Swietojanski et al., 2016] we are using the older set of Kaldi recipe scripts called `s5b`, and our baseline results are comparable with the corresponding baseline numbers previously reported. A newer set of improved scripts exists under `s5c` which, in comparison to `s5b`, offer about 1.5% absolute lower WER.

Table 4.4: %WER on Switchboard Hub00

System		`eval2000`		
Model	Features	SWB	CHE	TOTAL
GMM-HMM	LDA/STC+fMLLR	18.6	36.4	28.8
ANN-HMM	MFCC	15.8	28.4	22.1
ANN-HMM	LDA/STC	15.2	28.2	21.7

Table 4.5: % WER on Aurora 4. Clean and Multi-condition ANN models.

Model	A	B	C	D	AVG
ANN-HMM (clean)	4.8	26.2	21.2	43.5	31.7
ANN-HMM (multi-condition)	5.1	9.3	9.3	20.8	13.9

4.5 Aurora4

The Aurora 4 task is a small scale, medium vocabulary noise and channel ASR robustness task based on the Wall Street Journal corpus [Parihar et al., 2004]. We train our ASR models using the multi-condition training set. One half of the training utterances were recorded using a primary Sennheiser microphone, and the other half was collected using one of 18 other secondary microphones. The multi-condition set contains noisy utterances corrupted with one of six different noise types (airport, babble, car, restaurant, street traffic and train station) at 10-20 dB SNR. The standard Aurora 4 test set (`eval92`) consists of 330 utterances, which are used in 14 test conditions (4620 utterances in total). The same six noise types used during training are used to create noisy test utterances with SNRs ranging from 5-15dB SNR, resulting in a total of 14 test sets. These test sets are commonly grouped into 4 subsets – clean (group A, 1 test case), noisy (group B, 6 test cases), clean with channel distortion (group C, 1 test case) and noisy with channel distortion (group D, 6 test cases). We decode with the standard task's 5k bigram LM. The baseline results for clean and multi-condition training, following systems of Seltzer et al. [2013], are reported in Table 4.5.

Table 4.6: Hybrid system WER results on German eval set (geeval)

System description	Amount of training data		
	15hr	5hr	1hr
fBMMI+BMMI using LDA/STC	24.13	27.08	33.11
Tandem ML using LDA/STC	24.53	27.56	34.08
ANN-HMM	21.52	25.03	33.54

4.6 GlobalPhone multi-lingual corpus

The GlobalPhone corpus [Schultz, 2002] consists of recordings of speakers reading newspapers in their native language. There are 19 languages from a variety of geographic locations: Asia (Chinese, Japanese, Korean), Middle East (Arabic, Turkish), Africa (Hausa), Europe (French, German, Polish), and the Americas (Costa Rican Spanish, Brazilian Portuguese). There are about 100 speakers per language and about 20 hours of audio material. Recordings are made under relatively quiet conditions using close-talking microphones; however acoustic conditions may vary within a language and between languages.

Our setup is similar to that reported in Lu et al. [2014]. For unsupervised experiments we use German as our in-domain language and we simulated different degrees of available acoustic resources by selecting random 1 and 5 hour subsets of the total 15 hours of labeled training speech data.

We build standard maximum-likelihood (ML) trained GMM-HMM systems, using 39-dimensional MFCC features with delta and acceleration coefficients, on the full 15-hour training set for GlobalPhone German, as well as the 5-hour and 1-hour subsets. The number of context-dependent triphone states for the three systems are 2564, 1322 and 551, respectively, with an average of 16, 8 and 4 Gaussians, respectively, per state.

Our tandem systems use phone ANN posteriors obtained using a context window of 9 consecutive frames. We compare them with a baseline system where 9 frames (4 on each side of the current frame) of 13-dimensional MFCCs are spliced together and projected down to 40 dimensions using LDA and decorrelated with a single STC transform. We compare the hybrid setup to a GMM-HMM system that uses both model and feature-space discriminative training using boosted maximum mutual information estimation, referred to as fBMMI+BMMI in Table 4.6. The table also contains baseline tandem and ANN-HMM results.

Table 4.7: Statistics of the subset of GlobalPhone languages used in this work: the amounts of speech data for training, development, and evaluation sets are in hours.

Language	#Phones	#Spkrs	Train	Dev	Eval
German (DE)	41	77	14.9	2.0	1.5
Portuguese (PT)	45	101	22.8	1.6	1.8
Spanish (SP)	40	100	17.6	2.0	1.7
Swedish (SW)	52	98	17.4	2.0	-

4.7 Notes on statistical significance tests

If a test-set is small (say, less than an hour of speech) or if the WER difference between two systems is small, one may want to perform *statistical significance* tests in order to asses the likelihood of the hypothesis that the WERs are significantly different. In this work we will use the two-tailed Matched Pairs Sentence Segment Word Error (MAPSSWE) test [Gillick and Cox, 1989, Pallet et al., 1990], as implemented in the NIST Scoring Toolkit[11]. When we write in text "system A is better than system B", and do not explicitly state its significance level, 99.9% confidence level (p_v<0.001) is assumed. This follows recent recommendations for statistical testing [Johnson, 2013].

MAPSSWE testing requires pairwise comparisons between the systems of interest. It may, however, be of interest to be able to approximate this significance level by looking at the size of the particular test-set and the reference level of baseline errors [Povey, 2003]. We approximate the required WER change with respect to the assumed baseline level with the approach suggested by Bell [2010] (which is based on Povey [2003]), that is, under the assumption that the errors are independent one can model them with binomial distribution which for N sufficiently large can be approximated with normal distribution $\mathcal{N}(Np, Np(1-p))$. The proportion of interest of total errors has thus the mean p and variance $\frac{1}{N}p(1-p)$, where p is the probability of an error and N is a number of tokens in the test set. As such, the standard deviation of the error is thus $\sqrt{\frac{1}{N}p(1-p)}$. To get statistically significant bounds at the desired level one need to make sure the absolute change in error rates between two systems exceeds this standard deviation by 2, 2.6 or 3.3 for p_v<0.05, p_v<0.01 and p_v<0.001 confidence levels, respectively.

[11] http://www.nist.gov/itl/iad/mig/tools.cfm

Table 4.8: Approximations to significance levels for various test sets

Test set	Ref. #Tokens (N)	Baseline Ref. %WER ($100p/N$)	Abs. %WER change ($100\alpha\sqrt{\frac{p(1-p)}{N}}$) 95% $\alpha = 2$	99% $\alpha = 2.6$	99.9% $\alpha = 3.3$
`dev2010`	17511	15.5	0.55	0.71	0.90
`tst2010`	26994	15.0	0.43	0.56	0.72
`tst2011`	13699	12.0	0.55	0.73	0.92
`tst2013`	41720	22.0	0.41	0.52	0.67
`eval2000`	42637	16.0	0.36	0.46	0.59
`geeval`	11959	21.5	0.75	0.98	1.24
`aurora_eval92`	75474	14.0	0.25	0.33	0.42
`icsieval` (SDM)	36267	47.0	0.52	0.68	0.86
`amidev` (IHM/SDM)	108051	27.0 / 53.0	0.27 / 0.30	0.41 / 0.35	0.45 / 0.50
`amieval` (IHM/SDM)	102309	29.0 / 58.0	0.28 / 0.31	0.43 / 0.37	0.47 / 0.51

Those statistics for three significance levels are reported in Table 4.8.

It is worth mentioning that the approximations of both Povey [2003] and Bell [2010] are rather conservative and tend to under-estimate significance levels when compared to the MAPSSWE approach. We share this experience, for example, MAPSSWE statistical significance test performed on a control experiment using the TED `tst2010` set showed 0.1% absolute WER difference between two systems to be significant at p_v<0.006 level, while 0.2% WER difference is already significant at p_v<0.001. Contrast this with the required WER changes for `tst2010` in Table 4.8 where one would require 0.56% and 0.72% absolute WER change, respectively.

It is well known by ANN practitioners that the variance in the obtained results between different training sessions (initial seeds) may be significant. At the same time, it is computationally hard to derive error bars or uncertainty regions for ANN predictions for large-scale experiments. Recently this aspect has been more systematically studied for ASR by van den Berg et al. [2016] showing that standard deviations in WERs obtained with models estimated from reasonably large speech corpora (from 50 to 400 hours or so) can be substantial, reaching 0.27 – this is more than many improvements reported in the literature. Where possible, we try to repeat experiments to approximate the trust regions of our techniques.

Part II

Adaptation and Factorisation

Chapter 5

Learning Hidden Unit Contributions

The work in this chapter is an extended version of [Swietojanski, Li, and Renals, 2016] published in IEEE/ACM Transactions on Audio, Speech and Language Processing. Some further material is based on [Swietojanski and Renals, 2014] and [Swietojanski and Renals, 2016], published at IEEE Spoken Language Technology Workshop (SLT) and IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), respectively.

5.1 Introduction

Acoustic model adaptation [Woodland, 2001] aims to normalise the mismatch between training and runtime data distributions that arises owing to the acoustic variability across speakers, as well as other distortions introduced by the channel or acoustic environment. Initially adaptation techniques were primarily developed for GMM-HMM based ASR systems, although, it has been shown experimentally in many studies to date that adaptation of ANN acoustic models can also bring significant improvements in accuracy [Neto et al., 1995, Abrash et al., 1995, Li and Sim, 2010, Trmal et al., 2010, Yao et al., 2012, Yu et al., 2013b, Liao, 2013, Sainath et al., 2013c, Swietojanski et al., 2013a, Abdel-Hamid and Jiang, 2013] (see Sec. 5.2 for a more detailed overview).

Yu et al. [2013a] experimentally demonstrated that the invariance of ANN internal representations with respect to variabilities in the input space increases with depth (the number of layers) and that the ANN can interpolate well around training samples but fails to extrapolate if the data mismatch increases. Therefore one often explicitly compensates for unseen variabilities in the acoustic space.

This chapter is concerned with unsupervised model-based adaptation of ANN acoustic models to speakers and to acoustic environments, using a method termed *Learning Hidden Unit Contributions* (LHUC). We present the LHUC approach both in the context of test-only adaptation (Sec. 5.3), and extended to speaker-adaptive training (SAT) – SAT-LHUC (Sec. 5.4). We present an extensive experimental analysis using four standard corpora: TED talks, AMI, Switchboard and Aurora4. These experiments include and are organised as follows: adaptation of both cross-entropy and sequence trained ANN acoustic models (Sec. 5.5.1–5.5.3); an analysis in terms of the quality of adaptation targets, quality of adaptation data and the amount of adaptation data (Sec. 5.5.4); complementarity with feature-space adaptation techniques based on maximum likelihood linear regression (Sec. 5.5.5); and application to combined speaker and environment adaptation (Sec. 5.6).

5.2 Review of Neural Network Acoustic Adaptation

Approaches to the adaptation of neural network acoustic models can be considered as operating either in the feature space, or in the model space, or as a hybrid approach in which speaker-, utterance-, or environment-dependent auxiliary features are appended to the standard acoustic features. Those levels are visualised in Figure 5.1 and described below.

The dominant technique for estimating *feature space transforms* is constrained (feature-space) MLLR, referred to as fMLLR [Gales, 1998]. fMLLR is an adaptation method developed for GMM-based acoustic models, in which an affine transform of the input acoustic features is estimated by maximising the log-likelihood that the model generates the adaptation data based on first pass alignments. To use fMLLR with a ANN-based system, it is first necessary to train a complete GMM-based system, which is then used to estimate a single input transform per speaker. The transformed feature vectors are then used to train a ANN in a speaker adaptive manner and another set of transforms is estimated (using the GMM) during evaluation for unseen speakers. This technique has been shown to be effective in reducing WER across several different data sets, in both hybrid and tandem approaches [Mohamed et al., 2011, Seide et al., 2011, Hain et al., 2012, Hinton et al., 2012, Sainath et al., 2012, 2013c, Bell et al., 2013, Swietojanski et al., 2013a]. Similar techniques have also been developed to operate directly on neural networks. The linear input network (LIN) [Neto et al., 1995, Abrash et al.,

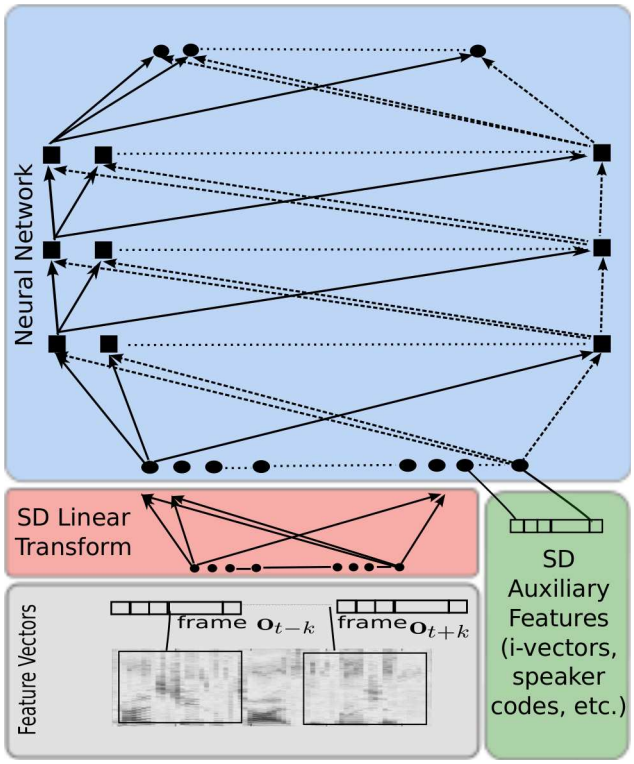


Figure 5.1: Illustration of adaptation levels of an ANN-HMM acoustic model.

1995] defines an additional speaker-dependent layer between the input features and the first hidden layer, and thus has a similar effect to fMLLR. This technique has been further developed to include the use of a tied variant of LIN in which each of the input frames is constrained to have the same linear transform [Li and Sim, 2010, Seide et al., 2011]. LIN and tied-LIN have been mostly used in test-only adaptation schemes; to make use of fMLLR transforms one needs to perform SAT training, which can usually better compensate against variability in acoustic space.

An alternative speaker-adaptive training approach – *auxiliary features* – augments the acoustic feature vectors with additional speaker-specific features computed for each speaker at both training and test stages. There has been considerable recent work exploring the use of i-vectors [Dehak et al., 2010] for this purpose. I-vectors, which can be regarded as basis vectors which span a subspace of speaker variability, were first used for adaptation in a GMM framework by Karafiat et al. [2011]. Saon et al. [2013] and Senior and Lopez-Moreno [2014] proposed to use i-vectors to augment the input features of ANN-based acoustic model and showed that appending (properly) normalised i-vectors for each speaker results in consistent accuracy gains. For example, Saon et al. [2013] re-

ports a 10% relative reduction in WER on Switchboard (and a 6% reduction on top, when the input features had been additionally transformed using fMLLR). Similar findings have been also found in [Gupta et al., 2014], while Karanasou et al. [2014] presented an approach in which the i-vectors were factorised into speaker and environment parts. Miao et al. [2015] proposed to transform i-vectors using an auxiliary ANN which produced speaker-specific transforms of the original feature vectors, similar to fMLLR. Other examples of auxiliary features include the use of speaker-specific bottleneck features obtained from a speaker separation ANN used in a distant speech recognition task [Liu et al., 2014], the use of out-of-domain tandem features [Bell et al., 2013], GMM-derived features [Tomashenko and Khokhlov, 2015] and speaker codes [Bridle and Cox, 1990, Abdel-Hamid and Jiang, 2013, Xue et al., 2014b] in which a specific set of units for each speaker is optimised. Speaker codes require speaker adaptive (re-)training, owing to the additional connection weights between codes and hidden units.

Model-based adaptation relies on a direct update of ANN parameters. Liao [2013] investigated supervised and unsupervised adaptation of different weight subsets using a few minutes of adaptation data. On a large net (60M weights), up to 5% relative improvement was observed for unsupervised adaptation when all weights were adapted. Yu et al. [2013b] have explored the use of regularisation for adapting the weights of a ANN, using the Kullback-Liebler (KL) divergence between the speaker-independent (SI) and speaker-dependent (SD) output distributions, resulting in a 3% relative improvement on Switchboard. This approach was also recently used to adapt all parameters of sequence-trained models [Huang and Gong, 2015]. One can also reduce the number of speaker-specific parameters through a different forms of factorisation. Examples include the use of singular value decomposition [Xue et al., 2014a] or built-in low-dimensional speaker transforms [Samarakoon and Sim, 2015]. Ochiai et al. [2014] have also explored regularised speaker adaptive training with a speaker-dependent (bottleneck) layer.

Directly adapting the weights of a large ANN results in extremely large speaker-dependent parameter sets, and a computationally intensive adaptation process. Smaller subsets of the ANN weights may be modified, including output layer biases [Yao et al., 2012], the bias and slope of hidden units [Zhao et al., 2015] or training the models with differentiable pooling operators, as we propose in Chapter 6, which are then adapted in SD fashion. Siniscalchi et al. [2013] also investigated the use of Hermite polynomial activation functions, whose pa-

rameters are estimated in a speaker adaptive fashion. One can also adapt the top layer in a Bayesian fashion resulting in a maximum a posteriori (MAP) approach [Huang et al., 2015b], or address the sparsity of context-dependent tied-states when few adaptation data-points are available by using multi-task adaptation, using monophones to adapt the context-dependent output layer, addressed in Chapter 7 [Swietojanski et al., 2015], [Huang et al., 2015a]. A similar approach, but using a hierarchical output layer (tied-states followed by monophones) rather than multi-task adaptation, has also been proposed [Price et al., 2014].

5.3 Learning Hidden Unit Contributions (LHUC)

A neural network may be viewed as a set of *adaptive basis functions*. Under certain assumptions on the family of target functions g^* (as well as on the model structure itself) the neural network can act as a universal approximator [Hornik et al., 1989, Hornik, 1991, Barron, 1993]. That is, given some vector of input random variables $\mathbf{x} \in \mathbb{R}^d$ there exists a neural network $g_n(\mathbf{x}) : \mathbb{R}^d \rightarrow \mathbb{R}$ of the form

$$g_n(\mathbf{x}) = \sum_{k=1}^{n} r_k \phi(\mathbf{w}_k^\top \mathbf{x} + b_k) \tag{5.1}$$

which can approximate g^* with an arbitrarily small error ϵ with respect to a distance measure such as mean square error (provided n is sufficiently large):

$$||g^*(\mathbf{x}) - g_n(\mathbf{x})||_2 \leq \epsilon. \tag{5.2}$$

In (5.1), $\phi : \mathbb{R} \rightarrow \mathbb{R}$ is an element-wise non-linear operation applied after an affine transformation which forms an adaptive basis function parametrised by a set of biases $b_k \in \mathbb{R}$ and weight vectors $\mathbf{w}_k \in \mathbb{R}^{d_\mathbf{x}}$. The target approximation may then be constructed as a linear combination of the basis functions, each weighted by $r_k \in \mathbb{R}$. The formulation can be extended to m-dimensional mappings $g_n^m(\mathbf{x}) : \mathbb{R}^d \rightarrow \mathbb{R}^m$ simply by splicing the models in (5.1) m times. The properties also hold true when considering deeper (nested) models [Hornik et al., 1989] (Corollaries 2.6 and 2.7).

ANN training results in the hidden units learning a joint representation of the target function and becoming specialised and complementary to each other. Generalisation corresponds to the learned combination of basis functions continuing to approximate the target function when applied to unseen test data. This interpretation motivates the idea of using `LHUC` – Learning Hidden Unit Contributions

– for test-set adaptation. In LHUC the network's basis functions, previously estimated using a large amount of training data, are kept fixed. Adaptation involves modifying the combination of hidden units in order to minimise the adaptation loss based on the adaptation data. Fig. 5.2 illustrates this approach for a regression problem, where the adaptation is performed by linear re-combination of basis functions changing only the r parameters from eq. (5.1).

The key idea of LHUC is to explicitly parametrise the amplitudes of each hidden unit (either in fully-connected or convolutional layers after max-pooling), using a speaker-dependent amplitude function. Let $h_j^{l,s}$ denote the j-th hidden unit activation (basis) in layer l, and let $r_j^{l,s} \in \mathbb{R}$ denote the s-th speaker-dependent amplitude parameter:

$$h_j^{l,s} = \xi(r_j^{l,s}) \circ \phi_j \left(\mathbf{w}_j^{l\top} \mathbf{x} + b_j^l \right). \tag{5.3}$$

The amplitude is re-parameterised using a function $\xi : \mathbb{R} \to \mathbb{R}^+$ – typically a sigmoid with range $(0, 2)$[1] [Swietojanski and Renals, 2014], but an identity function could be used [Zhang and Woodland, 2015]. $\mathbf{w}_j^l$ is the jth column of the corresponding weight matrix $\mathbf{W}^l$, b_j^l denotes the bias, ϕ is the hidden unit activation function (unless stated otherwise, this is assumed to be sigmoid), and $\circ$ denotes a Hadamard product[2]. ξ constrains the range of the hidden unit amplitude scaling (compare with Fig. 5.2) hence directly affecting the adaptation transform capacity – this may be desirable when adapting with potentially noisy unsupervised targets (see Sec. 6.5.1). LHUC adaptation progresses by setting the speaker-specific amplitude parameters $r_j^{l,s}$ using gradient descent with targets provided by the adaptation data.

The idea of directly learning hidden unit amplitudes was proposed in the context of an adaptive learning rate schedule by Trentin [2001], and was later applied to supervised speaker adaptation by Abdel-Hamid and Jiang [2013]. The approach was extended to unsupervised adaptation, non-sigmoid non-linearities, and large vocabulary speech recognition by Swietojanski and Renals [2014]. Other adaptive transfer function methods for speaker adaptation have also been proposed [Siniscalchi et al., 2013, Zhao et al., 2015], as have "basis" techniques [Wu

[1] This choice was initially motivated to have a constrained adaptation transform which is easy to plug into speaker independent model. The rationale is we want an initial gain of 1 for each hidden unit (corresponding to speaker-independent model) and then learning the importance of each hidden unit on adaptation data, but in limited manner. More analyses to follow.

[2] Although the equations are given in scalar form, we have used Hadamard product notation to emphasise the operation that would be performed once expanded to full-rank matrices.

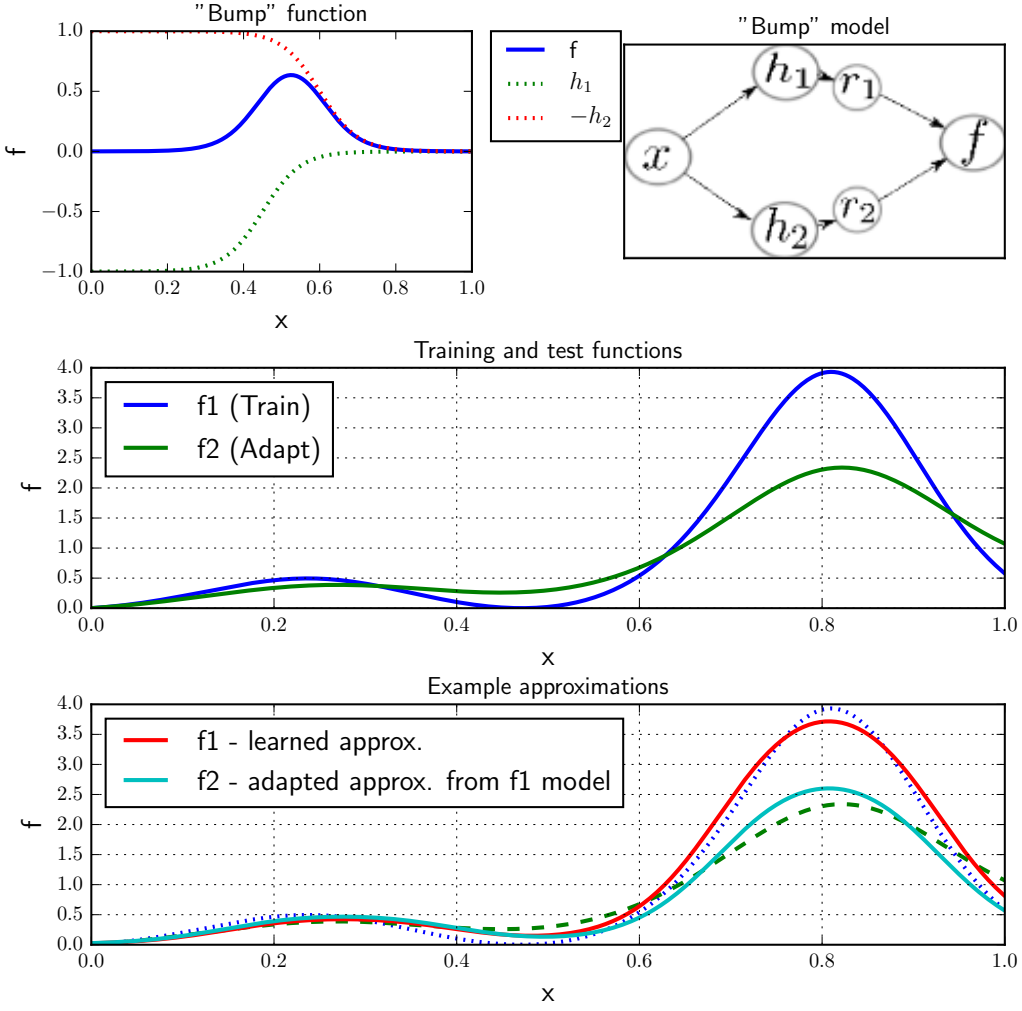


Figure 5.2: Example illustration of how LHUC performs adaptation (best viewed in color). Top: A "bump" model (5.1) with two hidden units can approximate "bump" functions. Middle: To learn function f_2 given training data f_1 (middle), we splice two "bump" functions together (4 hidden units, one input/output) to learn an approximation of function f_1. Bottom: LHUC adaptation of the model optimised to f_1 and adapted to f_2 using LHUC scaling parameters. Best viewed in color.

and Gales, 2015, Tan et al., 2015, Delcroix et al., 2015]. However, the basis in the latter works involved re-tuning parallel expert models on pre-defined clusters (gender, speaker, environment) in a supervised manner (somehow in the spirit of mixtures of experts by Jacobs et al. [1991]); the adaptation then relied on learning linear combination coefficients for those sub-models on adaptation data.

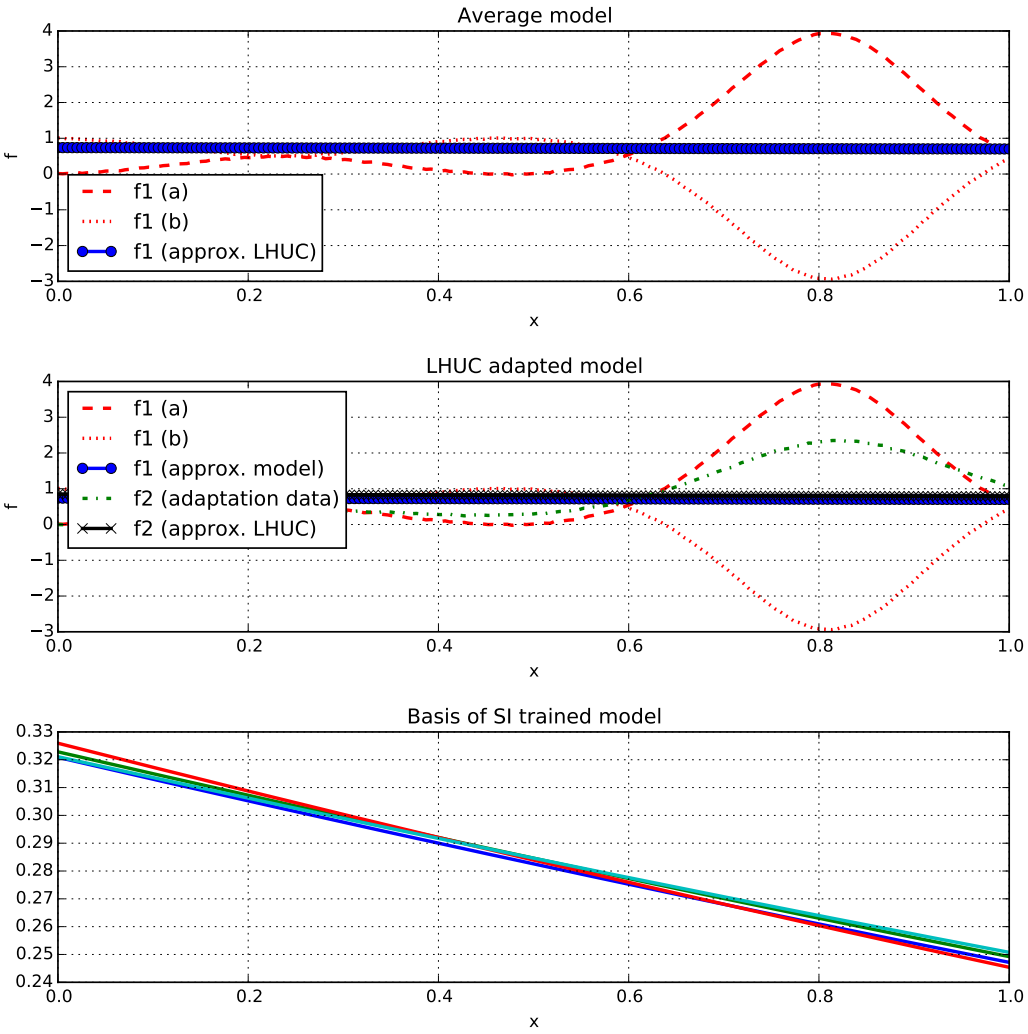


Figure 5.3: Top: A 4-hidden-unit model trained to perform regression on data drawn from two distributions $f1(a)$ and $f1(b)$ and the learned SI approximation optimal in the mean square error sense (blue). Middle: LHUC adapted representation to data $f2$ starting from the fixed SI basis functions (depicted in the bottommost plot) learned on $f1(a)$ and $f1(b)$. See text for more detailed description. Best viewed in color.

5.4 Speaker Adaptive Training LHUC (SAT-LHUC)

When LHUC is applied as a test-only adaptation it assumes that the set of speaker-independent basis functions estimated on the training data provides a good starting point for further tuning to the underlying data distribution of the adaptation data (Fig. 5.2). However, one can derive a counter-example where this assumption fails: the top plot of Fig. 5.3 shows example training data uniformly drawn from two competing distributions $f1(a)$ and $f1(b)$ where the linear recombination of the resulting basis in the average model (Fig 5.3 bottom), provides a poor approximation of adaptation data.

This motivates combining LHUC with speaker adaptive training (SAT) [Anastasakos et al., 1996] in which the hidden units are trained to capture both good average representations and speaker-specific representations, by estimating speaker-specific hidden unit amplitudes for each training speaker. This is visualised in Fig. 5.4 where, given the prior knowledge of which data-point comes from which distribution, we estimate a set of parallel LHUC transforms (one per distribution) as well as one extra transform which is responsible for modelling average properties. The top of Fig. 5.4 shows the same experiment as in Fig 5.3 but with three LHUC transforms – one can see that the 4-hidden-unit MLP in this scenario was able to capture each of the underlying distributions as well as the average aspect well, given the LHUC transform. At the same time, the resulting basis functions (Fig 5.4, bottom) are a better starting point for the adaptation (Fig. 5.4, middle).

The examples presented in Figs. 5.3 and 5.4 could be solved by breaking the symmetry through rebalancing the number of training data-points for each function, resulting in less trivial and hence more adaptable basis functions in the average model. However, as we will show experimentally later, similar effects are also present in high-dimensional speech data, and SAT-LHUC training allows more tunable canonical acoustic models to be built, that can be better tailored to particular speakers through adaptation.

Test-only adaptation for SAT-LHUC remains the same as for LHUC – the set of speaker-dependent LHUC parameters $\boldsymbol{\theta}^s_{LHUC} = \{r^{l,s}_j\}$ is inserted for each test speaker and their values optimised from unsupervised adaptation data. We also use a set of LHUC transforms $\boldsymbol{\theta}^s_{LHUC}$, where $s = 1 \dots S$, for the training speakers which are jointly optimised with the speaker-independent parameters $\boldsymbol{\theta}_{SI} = \{\mathbf{W}^l, \mathbf{b}^l\}$. There is an additional speaker-independent LHUC transform,

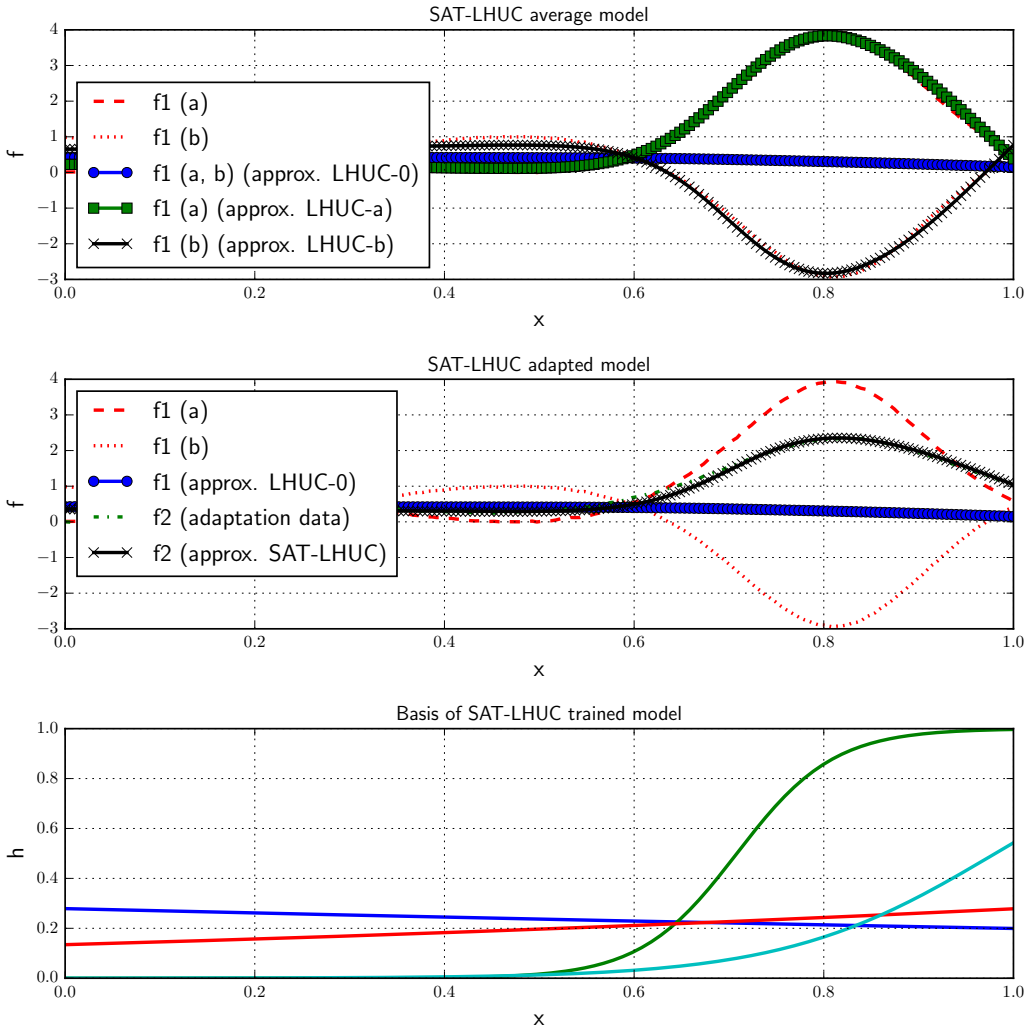


Figure 5.4: Learned solutions using three different SAT-LHUC transforms and shared basis functions: LHUC-0 learns to provide a good average fit to both distributions $f1(a)$ and $f1(b))$ at the same time, while LHUC-a and LHUC-b are tasked to fit either $f1(a)$ or $f1(b)$, respectively. The bottom plot shows the resulting basis functions (activations of 4 hidden units) of the SAT-LHUC training approach - one can observe SAT-LHUC provides a richer set of basis function which can fit the data well on average, and can also capture some underlying characteristics necessary to reconstruct target training data – using different LHUC transforms, this property is also visualised in the middle plot. (Best viewed in color.)

denoted by $\boldsymbol{\theta}^0_{LHUC}$, which allows the model to be used in speaker-independent fashion, for example, to produce first pass adaptation targets. This joint learning process of hidden units with speaker-dependent `LHUC` scaling is important, as it results in a more tunable canonical acoustic model that can be better adjusted to unseen speakers at test time, as illustrated in Fig. 5.4 and demonstrated on adaptation tasks in the following sections.

To perform SAT training with `LHUC`, we use cross-entropy objective to maximise the posterior probability of obtaining the correct context-dependent tied-state q_t given observation vector $\bar{\mathbf{O}}_t$ at time t:

$$\mathcal{F}_{SAT}(\boldsymbol{\theta}_{SI}, \boldsymbol{\theta}_{SD}) = -\sum_{t \in D} \log P(q_t | \bar{\mathbf{O}}^s_t; \boldsymbol{\theta}_{SI}; \boldsymbol{\theta}^{m_t}_{LHUC}) \tag{5.4}$$

where s denotes the sth speaker, $m_t \in \{0, s\}$ selects the SI or SD `LHUC` transforms from $\boldsymbol{\theta}_{SD} \in \{\boldsymbol{\theta}^0_{LHUC}, \ldots, \boldsymbol{\theta}^S_{LHUC}\}$ based on a Bernoulli distribution:

$$k_t \sim \text{ Bernoulli}(\gamma) \tag{5.5}$$

$$m_t = \begin{cases} s & \text{if } k_t = 0 \\ 0 & \text{if } k_t = 1 \end{cases} \tag{5.6}$$

where γ is a hyper-parameter specifying the probability that the given example is treated as SI. The SI/SD split (determined by (5.5) and (5.6)) can be performed at speaker, utterance or frame level. We further investigate this aspect in Section 5.5.2. The `SAT-LHUC` model structure is depicted in Fig 5.5; notice the alternative routes of ANN training forward and backward passes for different speakers.

Denote by $\partial \mathcal{F}_{SAT} / \partial h^{l,s}_j$ the error back-propagated to the jth unit at the lth layer (5.3). To back propagate through the transform one needs to element-wise multiply it by the transform itself, as follows:

$$\frac{\partial \mathcal{F}_{SAT}}{\partial \psi^l_j} = \frac{\partial \mathcal{F}_{SAT}}{\partial h^{l,s}_j} \circ \xi(r^{l,s}_j) \tag{5.7}$$

To obtain the gradient with respect to $r^{l,s}_j$:

$$\frac{\partial \mathcal{F}_{SAT}}{\partial r^{l,s}_j} = \frac{\partial \mathcal{F}_{SAT}}{\partial h^{l,s}_j} \circ \frac{\partial \xi(r^{l,s}_j)}{\partial r^{l,s}_j} \circ \psi^l_j \tag{5.8}$$

When performing mini-batch SAT training one needs to explicitly take account of the fact that different data-points (indexed by t in (5.9)) may flow through

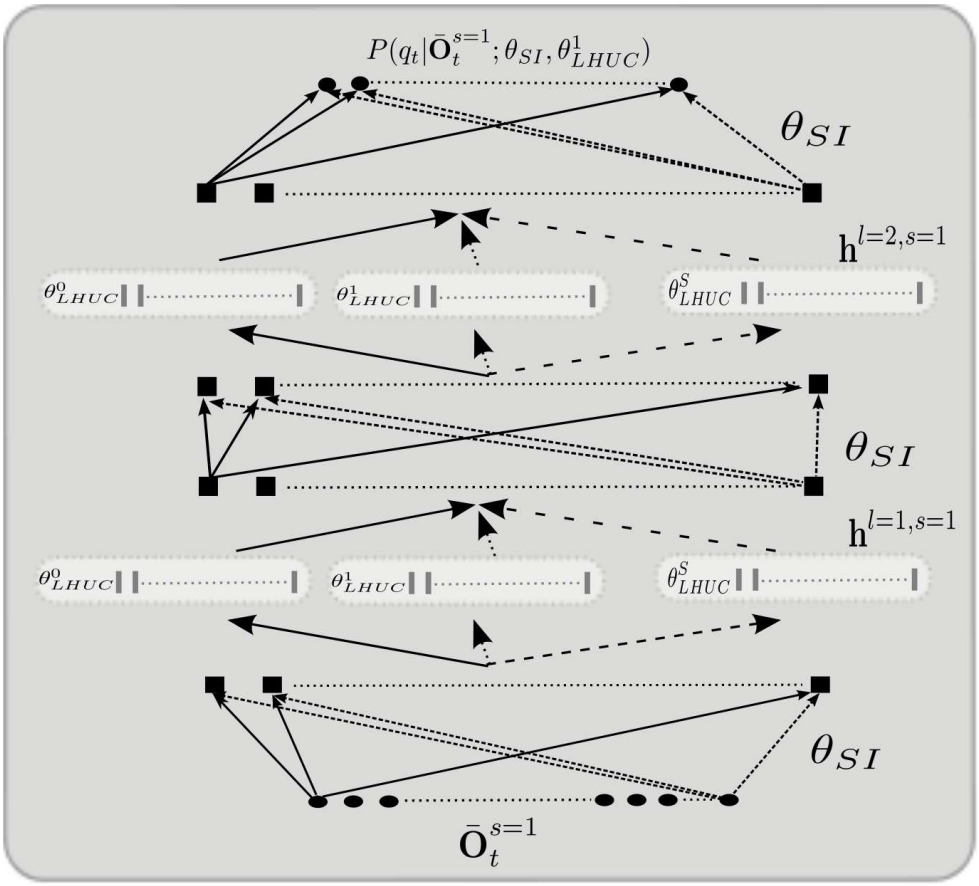


Figure 5.5: Schematic of SAT-LHUC training, with a data point from speaker $s = 1$. Dashed line indicates an alternative route through the SI LHUC transform.

different transforms: hence the resulting gradient for $r_j^{l,s}$ for the sth speaker is the sum of the partial gradients belonging to speaker s (or SI LHUC transform):

$$\frac{\partial \mathcal{F}_{SAT}}{\partial r_j^{l,s}} = \sum_{t,m_t==s} \frac{\partial \mathcal{F}_{SAT}}{\partial h_j^{l,s}} \circ \frac{\partial \xi(r_j^{l,s})}{\partial r_j^{l,s}} \circ \psi_j^l \tag{5.9}$$

or 0 in case no data-points for sth speaker in the given mini-batch were selected. All adaptation methods studied in this chapter require first-pass decoding to obtain adaptation targets to either estimate fMLLR transforms for unseen test speakers or to perform ANN speaker-dependent parameter update.

5.5 Results

We experimentally investigated LHUC and SAT-LHUC using four different corpora: the TED talks corpus (Section 4.2); the Switchboard corpus of conversational telephone speech (Section 4.4); the AMI meetings corpus (Section 4.3.1); and the Aurora4 corpus of read speech with artificially corrupted acoustic environments (Section 4.5). Unless explicitly stated otherwise, the models share similar structure across the tasks – ANNs with 6 hidden layers (2,048 units in each) using a sigmoid non-linearity. The output logistic regression layer models the distribution of context-dependent clustered tied states [Dahl et al., 2012]. The features are presented in 11 ($\pm$5) frame long context windows. All the adaptation experiments, unless explicitly stated otherwise, were performed unsupervised.

5.5.1 LHUC hyperparameters

Our initial study concerned the hyper-parameters used with LHUC adaptation. First, we used the TED talks to investigate how the word error rate (WER) is affected by adapting different layers in the model using LHUC transforms. The results, graphed in Fig. 5.6 (a), indicated that adapting only the bottom layer brings the largest drop in WER; however, adapting more layers further improves the accuracy for both LHUC and SAT-LHUC approaches (adapting the other way round – starting from the top layer – is much less effective). Since obtaining the gradients for the r parameters at each layer is inexpensive compared to the overall back-propagation, and we want to adapt at least the bottom layer, we apply LHUC to each layer for the rest of this work.

Table 5.1: WER(%) for different re-parametrisation functions for LHUC transforms on TED tst2010. Unadapted baseline WER is 15.0%.

r	$2/(1+\exp(-r))$	$\exp(r)$	$\max(0,r)$
12.8	12.8	12.7	12.7

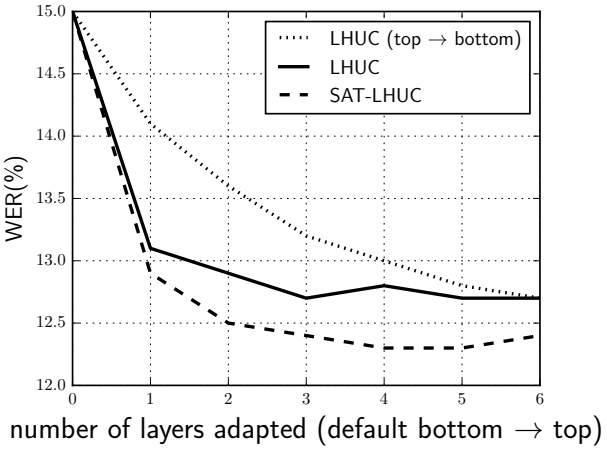
Fig. 5.6 (b) shows how WERs varies with the number of adaptation iterations. The results indicate that one sweep over the adaptation data (in this case tst2010) is sufficient and, more importantly, the model does not overfit when adapting with more iterations (despite frame error rates consistently decreasing – Fig. 5.6 (c)). This suggests that it is not necessary to carefully regularise the model – for example, by Kullback-Leibler divergence training [Yu et al., 2013b] which is required when adapting the weights of one or more layers in a network.

Finally, we explored how the form of the LHUC re-parametrisation function ξ affects the WER and frame error rate (FER) (Fig. 5.6 (c) and Table 5.1). For test-only adaptation only a small WER difference (0.1% absolute, statistically insignificant) is observed, regardless of the large difference in frame accuracies. This supports our previous observation that LHUC is robust against over-fitting. For SAT-LHUC training, a less constrained parametrisation was found to give better WERs for the SI model. Based on our control experiments, during SAT-LHUC training, setting ξ to be the identity function (linear r) gave similar results to $\xi(r) = \max(0, r)$ and $\xi(r) = \exp(r)$ and all were better than re-parametrising with $\xi(r) = 2/(1 + \exp(-r))$. This is expected as for full training the last approach constrains the range of back-propagated gradients. From now on, if not stated otherwise, we will use $\xi(r) = \exp(r)$.

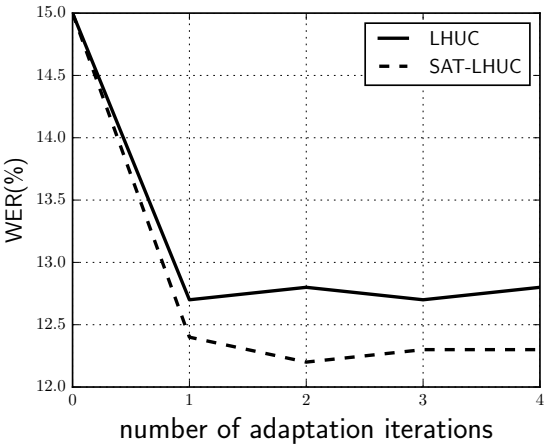
We adapt all our models with the learning rate set to 0.8 (regardless of $\xi(\cdot)$) and the basic training of both the SI and the SAT-LHUC models was performed with the initial learning rate set to 0.08 and was later adjusted according to the newbob learning scheme [Renals et al., 1992].

5.5.2 SAT-LHUC

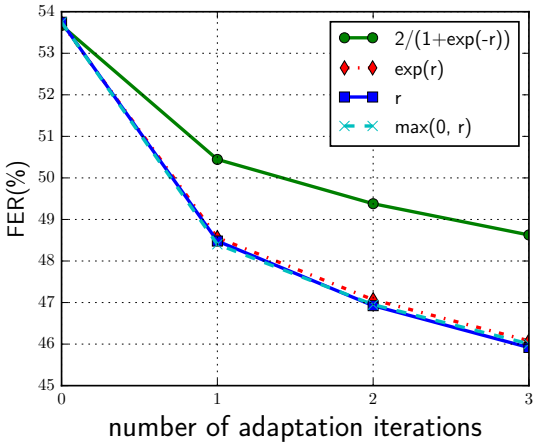
As described in Section 5.4, SAT-LHUC training aims to encourage the hidden unit feature receptors so that they capture not just the average characteristics of training data, but also specific features of the different distributions the data was drawn from (for example, different training speakers). As a result, the model can



(a)



(b)



(c)

Figure 5.6: WER(%) on TED `tst2010` as a function of: (a) number of adapted layers (3 adaptation iterations) and (b) number of adaptation iterations (all hidden layers adapted), (c) FER (%) for re-parameterisation functions (ξ) used in adaptation.

Table 5.2: WER(%) on TED `tst2013` for different sampling strategies and `SAT-LHUC` training and adaptation. Baseline system is adapted with test-only `LHUC`. See Table 5.9 for more comparisons with other adaptation techniques and TED test-sets.

		WER (%) for sampling strategies		
Model	Baseline	Per Speaker	Per Segment	Per Frame
SI	22.1	23.0	22.0	22.0
SD	19.1	18.6	18.1	18.0

be better tailored to unseen speakers by putting more importance to those units that were useful for training speakers with similar characteristics.

Prior to `SAT-LHUC` training we need to decide on how and which data should be used to estimate speaker-dependent and speaker-independent transforms[3]. In this thesis we investigate `SAT-LHUC` models with frame-level, segment-level and speaker-level clusters. For speaker- and segment-level transforms we decide which speakers or segments are going to be treated as SI or SD prior to training (and this choice is kept fixed through all training epochs). For the frame-level `SAT-LHUC` approach, the SI/SD decisions are made separately for each data-point during training. In either scenario we ensure that the overall SD/SI ratio determined by γ parameter is satisfied. The WER results for each of these three approaches ($\gamma = 0.5$) are reported in Table 5.2 and Figure 5.7 for several settings of γ. Speaker-level `SAT-LHUC` training provides the highest WERs for both SI and SD decodes (0.6% abs. above the one obtained at frame-level, p_v<0.001). Segment-level and frame-level `SAT-LHUC` training result in similar WERs for SI decodes, with a small advantage (0.1% abs., statistically insignificant) for the frame-level approach after adaptation.

We then investigate the impact of the SI/SD ratio when training the ANN weights and the SI and SD `LHUC` transforms. The SI/SD ratio depends on γ, the hyper-parameter in eq (5.5). To speed-up the experimental turnaround we initially limited our experiments to the TED corpus with 30 hours training data, using smaller models (1,000 hidden units per layer). The segments for this limited condition were sampled in such a way that the number of speakers remained the

[3]At double cost one could treat each example as both SI and SD propagating it forward and backward twice through the network, and then compute two sets of the corresponding gradients for the parameters. In this thesis we follow an approach which keeps training time constant by introducing an additional hyper-parameter that specifies the ratio of SI and SD data

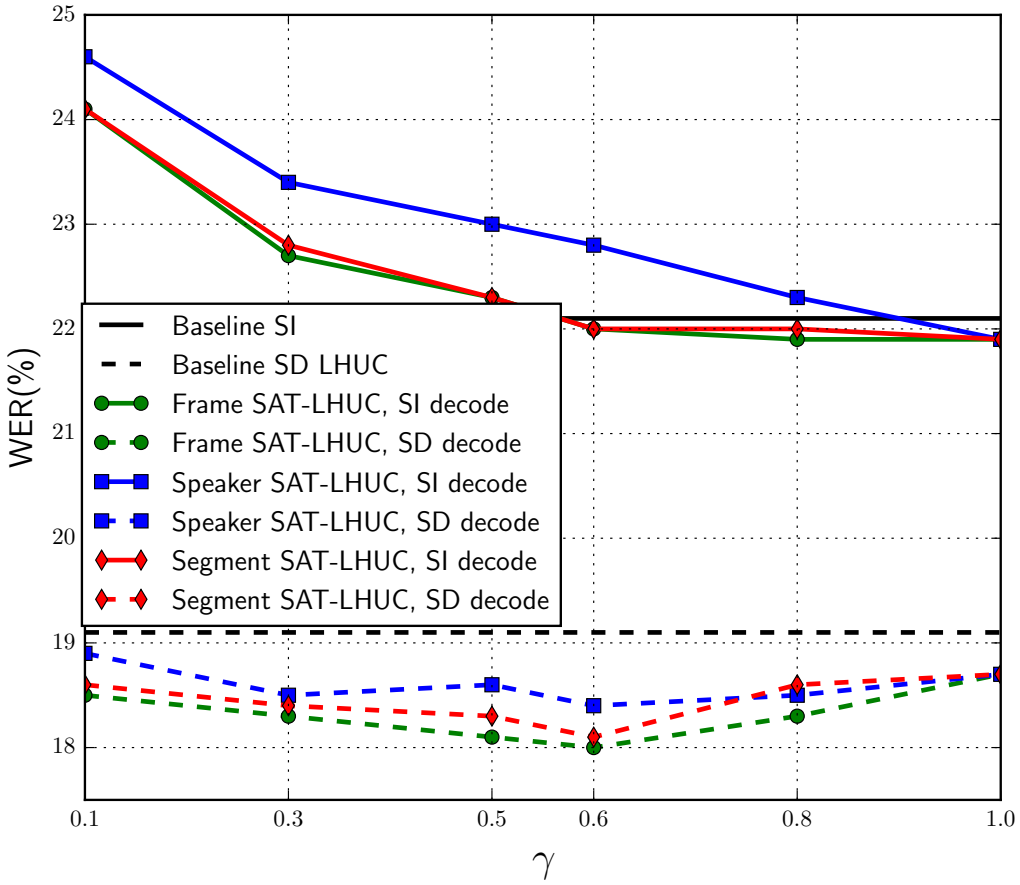


Figure 5.7: WER(%) for different sampling strategies {per frame, per segment, per speaker} for SAT-LHUC training and SI and SD decodes on TED `tst2013`.

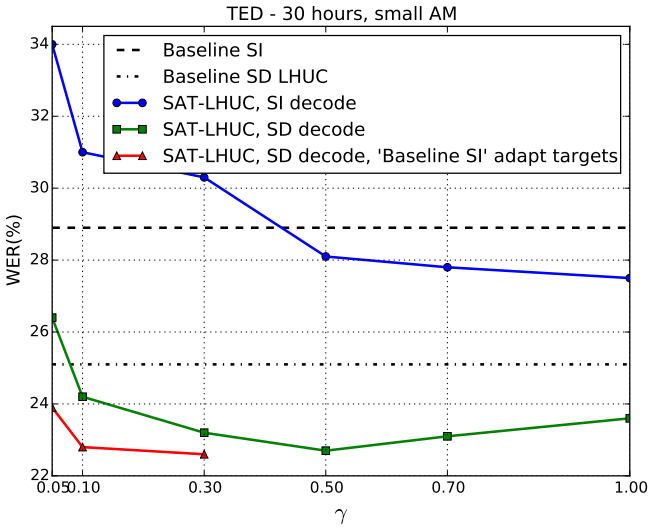
same between the limited and full variants. Results of those experiments on `tst2013`, for different settings of $\gamma \in \{0.05, 0.1, 0.3, 0.5, 0.7, 1.0\}$, can be found in Fig. 5.8 (a). Note, when $\gamma = 0$ the SI transform would not be estimated; conversely for $\gamma = 1.0$ there would be only a single global SI transform. The latter case is a variant of parametrised sigmoid activations with a learnable amplitude during training [Zhang and Woodland, 2015].

The first observation one can draw from Fig. 5.8 is that the accuracy of the `SAT-LHUC` model and the SI decodes depends on the amount of data used to estimate the SI `LHUC` transforms during training – the less SI data that flows through SI `LHUC` transforms, the worse SI results are, with a dramatic decrease in first-pass accuracy when less than 30% of data is treated as speaker-independent ($\gamma < 0.3$). Conversely, increasing the SI/SD ratio to about 50% results in comparable accuracy to the standalone SI-trained model. This trend holds for other scenarios with more data, including Full-TED (i.e. 143 hours training data) (Fig. 5.8 (b)) and SWBD (Fig. 5.8 (c)).

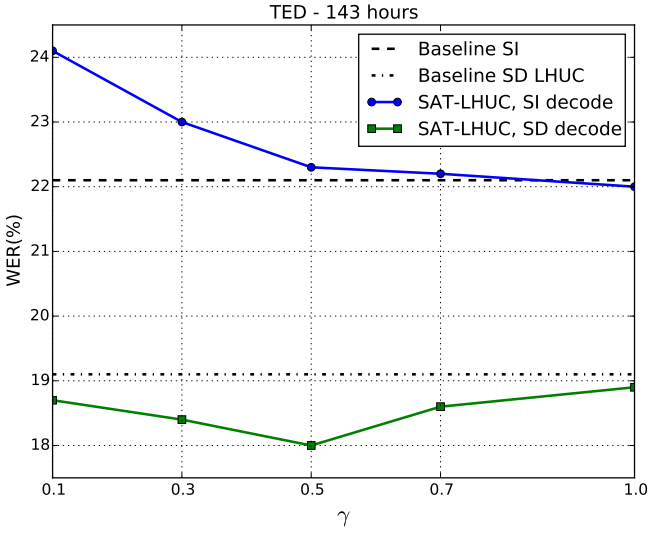
The parametrised sigmoid function (for $\gamma = 1.0$) is particularly effective for data-constrained experiments (compare Fig. 5.8 (a) with (b) and (c)); for instance, on 30hour-TED the parametrised sigmoid model results in a WER of 27.5% while the conventional sigmoid model has a WER of 28.9%. This advantage diminishes for bigger models and more data.

Then we investigated how `SAT-LHUC` affects the accuracy of `LHUC` adapted systems. To do so we adapted `SAT-LHUC` models using the first pass adaptation targets obtained from the corresponding SAT-LHUC systems operating in SI mode. Here we can see that a speaker-dependent representation provides a more tunable canonical model. For example, on 30hour-TED an adapted `SAT-LHUC` $\gamma = 0.3$ system produced 8% relative lower WERs when compared to an adapted SI system (23.2% vs. 25.1%), regardless of the fact that the `SAT-LHUC` adaptation alignments were 1.4% absolute worse than its SI counterpart (30.3% vs. 28.9%). Those results are also reported in Table 5.2.

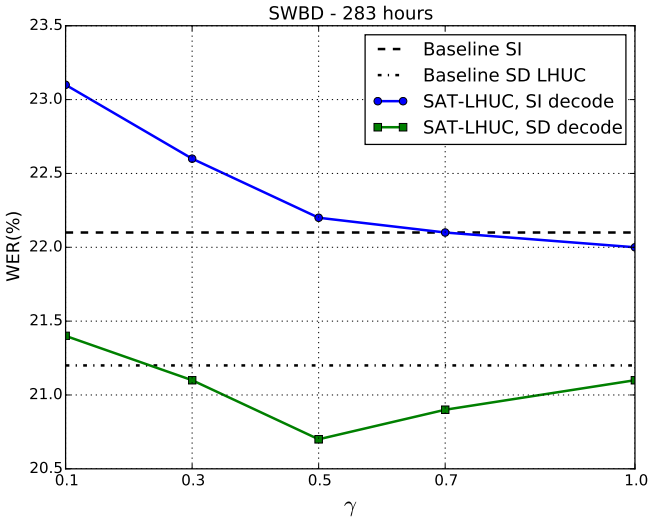
Finally, we investigated whether the inferior adaptation results for $\gamma < 0.3$ were caused by differences in learned representations or by lower quality adaptation targets. We used the adaptation targets of the 'Baseline SI' model (28.9% WER) and adapted `SAT-LHUC` models trained with $\gamma \in \{0.05, 0.1, 0.3\}$ on 30hour-TED. The results (Fig. 5.8 (a)) indicate that the reason for lower adaptation accuracies (compared to $\gamma = 0.5$ system) was mostly due to less accurate adapta-



(a) `tst2013` using 30hour-TED AM



(b) `tst2013` using Full-TED AM



(c) `eval2000` using SWBD MFCC AM

Figure 5.8: WER(%) as a function of γ in equation (5.5).

Table 5.3: WER(%) on TED talks (`tst2010` and `tst2013`).

System		IWSLT Test set	
Training	Decoding	`tst2010`	`tst2013`
Baseline speaker-independent systems			
SI	SI	15.0	22.1
SAT-LHUC	SI	15.1	22.0
Adapted systems			
SI	`LHUC`	12.7	19.1
`SAT-LHUC`	`LHUC`	12.4	18.0

tion targets. Adapting the $\gamma = 0.3$ model with the 'Baseline SI' targets reduces the WERs of $\gamma = 0.3$ system to 22.6% (from 23.2%) – 2.5% absolute lower when compared the baseline SD LHUC system (25.1%) (both systems used the same adaptation targets) and 0.1% absolute lower than the best $\gamma = 0.5$ system. This further strengthens our claim that the `SAT-LHUC` models indeed learn a better and more tunable speaker-dependent representation, but their use is somehow limited by the necessary trade-off of managing a good SI first-pass model. If different models for SI and SD decodes are acceptable, then further small gains in accuracy are observed as discussed earlier and shown in Figure 5.8 (a).

Fig. 5.8 (c) shows similar plot but for Switchboard data (more detailed discussion below) and one can observe a similar pattern, with $\gamma = 0.5$ being an optimal choice. This, in conjunction with another validation on AMI data, is a strong indicator that SAT-LHUC training with roughly half of the data-points being treated as speaker-independent makes a good task-independent setting.

We report the baseline `LHUC` and `SAT-LHUC` comparisons on TED and AMI data in Tables 5.3 and 5.4, respectively (further results, including a comparison to fMLLR transforms and on Switchboard data are in the next sections). On TED (Table 5.3), `SAT-LHUC` models operating in SI mode ($\gamma = 0.6$) have comparable WERs to SI models; however, adaptation resulted in a WER reduction of 0.3–1.1% absolute (2–6% relative) compared to test-only adaptation of the SI models. Similar results were observed on the AMI data (Table 5.4) where for both ANN and CNN models trained on FBANK features `LHUC` adaptation decreased the WER by 2% absolute (7% relative) and `SAT-LHUC` training improved this result by 4% relative for ANN models. As expected, the `SAT-LHUC` gain for CNNs was smaller when compared to ANN models, since the CNN layer can learn different

Table 5.4: WER(%) on AMI–IHM

Model	amidev	amieval
ANN	26.8	29.1
+LHUC	25.6	27.1
+SAT-LHUC	24.9	26.1
CNN	25.2	27.1
+LHUC	24.3	25.3
+SAT-LHUC	23.9	24.8

patterns for different speakers which may be selected through the max-pooling operator at run-time. We further exploit this aspect by proposing the use of differentiable pooling operators for adaptation in Chapter 6.

5.5.3 Sequence model adaptation

Model-based adaptation of sequence-trained ANNs (SE-ANN) is more challenging compared to adapting networks trained using cross-entropy: a mismatched adaptation objective (here cross-entropy) can easily erase sequence information from the weight matrices due to the well-known effect of catastrophic forgetting [French, 1999] in neural networks. Indeed Huang and Gong [2015] report no gain from adapting SE-ANN models with a cross-entropy adaptation objective and supervised adaptation targets. In those experiments, all weights in the model were updated and KL divergence regularised adaptation [Yu et al., 2013b] or KL regularised sequence level adaptation were required to further improve on the SE-ANN. It remains to be answered if one can obtain similar improvements using SE-ANN adaptation and first-pass transcripts.

In this work we adapt state-level minimum Bayes risk (sMBR) [Kaiser et al., 2000, Povey, 2003, Kingsbury, 2009] sequence-trained models using LHUC and report results on TED tst2011 and tst2013 in Table 5.5. We kept all the LHUC adaptation hyper-parameters the same as for CE models and obtained around 2% absolute (11% relative) WER reductions on tst2013 for both SI and fMLLR SAT adapted SE-ANN systems. Interestingly, the obtained adaptation gain was similar to the cross-entropy models and LHUC adaptation did not seem to disrupt the learned model's sequence representation.

We compared our adaptation results to the most accurate system of the

Table 5.5: Summary of WER results of LHUC adapted sequence models on TED tst2011 and tst2013

Model	tst2011	tst2013
ANN-CE	12.1	22.1
ANN-sMBR	10.3	20.2
+LHUC	9.5	18.0
+fMLLR	9.6	18.9
++LHUC	8.9	15.8

IWSLT–2013 TED transcription evaluation, which performed both feature- and model-space speaker adaptation [Huang et al., 2013a]. For model-space adaptation that system used a method which adapts ANNs with a speaker-dependent layer [Ochiai et al., 2014]. The results are reported in Table 5.6 where in the first block one can see a standard sequence-trained feature-space adapted system build from TED and 150 hours of out-of-domain data scoring 15.7% WER, similar to the WER of our TED system (15.4%), which also for IWSLT utilised 100 hours of out-of-domain AMI data. The 0.3% difference could be explained by characteristics of the out-of-domain data used (tst2013 is characterised by a large proportion of non-native speakers which is also typical for AMI data, hence benefits more our baseline systems). When comparing both adaptation approaches operating in an unsupervised manner one can see that LHUC gives much bigger improvements in WER compared to speaker-dependent layer, 2.1% vs. 0.6% absolute (14% vs. 4% relative) on tst2013. This allows our single-model system to match a considerably more sophisticated post-processing pipeline of Huang et al. [2013a], as outlined in Table 5.6. For less mismatched data (tst2011) adaptation is less important and our system has a WER 0.8% absolute higher compared with the more sophisticated system.

From these experiments we conclude that LHUC is an effective way to adapt sequence models in an unsupervised manner using a cross-entropy objective function, without the risk of removing learned sequence information. It is to be seen if unsupervised adaptation of LHUC scalers with sMBR criterion remains effective (in the spirit of the work of Huang and Gong [2015]).

Table 5.6: WERs for adapted sequence-trained models used in IWSLT evaluation. Note, the results are not directly comparable to those reported on TED in Table 5.5 due different training data and feature pre-processing pipelines (see referenced papers for system details).

Model	tst2011	tst2013
IWSLT2013 winner system (numbers taken from [Huang et al., 2013a])		
ANN (sMBR) + HUB4 + WSJ	-	15.7
+ Six ROVER subsystems	-	14.8
++ Automatic segmentation	-	14.3
+++ LM adapt. + RNN resc.	-	14.1
+++++ SAT on ANN [Ochiai et al., 2014]	7.7	13.5
Our system [Bell et al., 2014]		
ANN (sMBR) + AMI data	9.0	15.4
+LHUC	8.5	13.3

5.5.4 Other aspects of adaptation

5.5.4.1 Amount of adaptation data

Fig 5.9 shows the effect of the amount of adaptation data on WER for LHUC and SAT-LHUC adapted models. As little as 10s of unsupervised adaptation data is already able to substantially decrease WERs (by 0.5–0.8% absolute). The improvement for SAT-LHUC adaptation compared with LHUC is considerably larger – roughly by a factor of two up to 30s adaptation data. As the duration of adaptation data increases the difference gets smaller; however SAT-LHUC results in consistently lower WERs than LHUC in all cases (including full two pass adaptation).

We also investigated supervised (oracle) adaptation by aligning the acoustics with the reference transcriptions (dashed lines). Given supervised adaptation targets, LHUC and SAT-LHUC further substantially decrease WERs, with SAT-LHUC giving a consistent advantage over LHUC.

5.5.4.2 Quality of adaptation targets

Since our approach relies on a two-pass decoding, we investigated the extent to which LHUC is sensitive to the quality of the adaptation targets obtained in the first-pass recognition pass. In this experiment we explored the differences

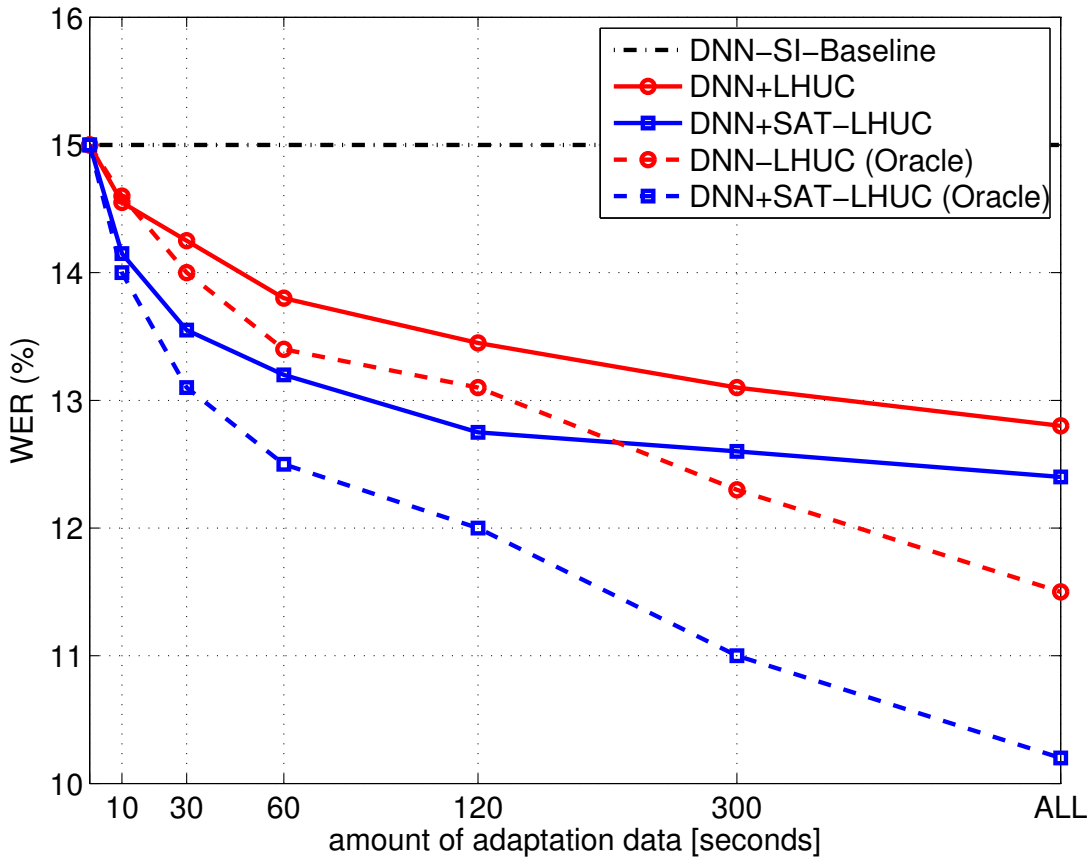


Figure 5.9: WER(%) for unsupervised and oracle adaptation data on TED `tst2010`.

resulting from different language models, and assumed that the first pass adaptation data was generated by either an SI or a SAT-LHUC model operating in SI mode. The main results are shown in Fig 5.10 where the solid lines show WERs obtained with a pruned 3-gram LM (3gm-312MW.p07) and different types of adaptation targets resulting from re-scoring the adaptation data with stronger LMs (3gm-312MW, 4gm-312MW, 4gm-751MW– see Section 4.2 for detailed description). One can see there is not much difference unless the adaptation data was re-scored with the largest 4-gram LM (4gm-751MW). This improvement diminishes in the final adapted system after re-scoring. This suggests that the technique is not very sensitive to the quality of adaptation targets. This trend holds regardless of the amount of data used for adaptation (ranging from 10s to several minutes per speaker). In related work [Miao et al., 2015] LHUC was employed using alignments obtained from an SI-GMM system with a 8.1% absolute higher WER than the corresponding SI ANN, and substantial gains were obtained over the unadapted SI ANN baseline – although the WER reduction was considerably smaller (1% absolute) compared to adaptation with alignments obtained with the corresponding SI ANN.

5.5.4.3 Quality of data

We also investigated how the quality of the acoustic data itself affects the adaptation accuracies, keeping the other ASR components fixed. We performed an experiment on the AMI corpus using speech captured by individual headset microphones (IHM) and a single distant tabletop microphone (SDM). In case of IHM we adapt to the headset; in this experiment we assume we have speaker labels for the SDM data[4]. The results are reported in Table 5.7: LHUC adaptation improves the accuracy in both experiments, although the gain for the SDM condition is smaller; however, the SDM system is characterised by twice as large WERs. Notice that LHUC has also been successfully applied to channel normalisation between distant and close talking microphones [Himawan et al., 2015].

5.5.4.4 One-shot adaptation

By one-shot adaptation we mean the scenario in which LHUC transforms were estimated once for a held-out speaker and then used many times in a single

[4]In a real scenario for SDM data one would have to perform speaker diarisation in order to obtain speaker labels.

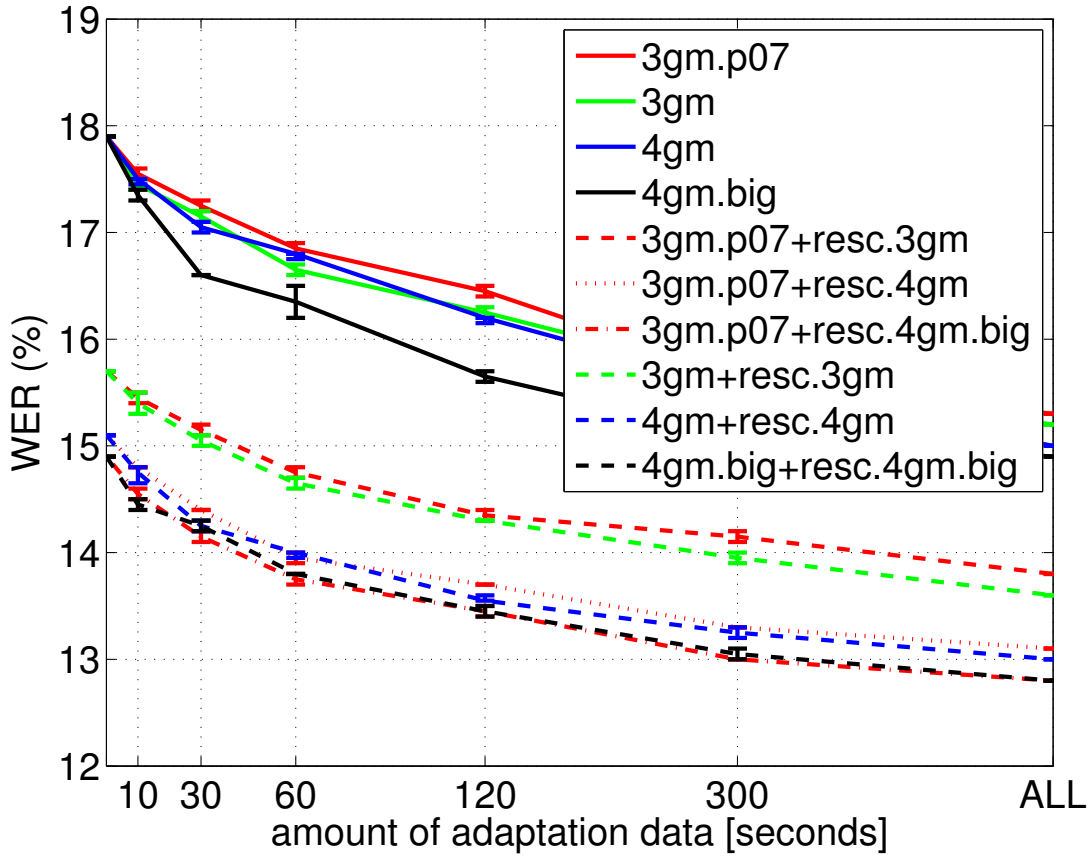


Figure 5.10: WER(%) for different qualities of adaptation targets on TED `tst2010`. Terms 3gm.p07, 3gm, 4gm and 4gm.big correspond to `3gm-312MW.p07`, `3gm-312MW`, `4gm-312MW` and `4gm-751MW`, respectively. Pruning threshold was set to 10^{-7}. Notation 'LM+resc.LM' means the first pass adaptation targets were obtained with 'LM' (possibly re-scored with this LM) and the corresponding adapted model second pass hypotheses were finally re-scored with 'resc.LM'. See text for further description.

Table 5.7: WER(%) on AMI–IHM and AMI–SDM using adapted CNNs.

Model	amidev	amieval
CNN (IHM)	25.2	27.1
+LHUC	24.3	25.3
CNN (SDM)	49.8	54.4
+LHUC	48.8	53.1

Table 5.8: WER(%) on AMI–IHM and one-shot adaptation

Model	amidev	amieval
CNN	25.2	27.1
+LHUC	24.3	25.3
+LHUC.one-shot	24.3	25.4

pass system for this speaker. We performed those experiments on AMI IHM data, and report results on amidev and amieval which contain 21 and 16 unique speakers taking part in 18 and 16 different meetings, respectively. Each speaker participates in multiple meetings: to some degree, adapting to a speaker in one meeting, then applying the adaptation transform to the same speaker in the other meetings simulates a real-life condition where it is possible to assume the speaker identity without having to perform speaker diarisation (e.g. personal devices). The results of this experiment (Table 5.8) indicate that LHUC retains the accuracies of two-pass systems by providing almost identical results when comparing LHUC estimated in a full two-pass system and when the unsupervised transforms are re-used in the LHUC.one-shot experiment.

5.5.5 Complementarity to feature normalisation

Feature-space adaptation using fMLLR is a very reliable current form of speaker adaptation, so it is of great interest to explore how complementary the proposed approaches are to SAT training with fMLLR transforms. In this work we do not make any explicit comparisons to other techniques such as auxiliary i-vector features or speaker-codes; however, the literature suggest that the use of i-vectors give similar [Saon et al., 2013] results when compared to fMLLR trained models. Related recent studies also show LHUC is at least as good as the standard use of i-vector features [Miao et al., 2015, Samarakoon and Sim, 2016].

We compared LHUC and SAT-LHUC to SAT-fMLLR training using TED tst2010 (Fig 5.11, red curves). We also compared both techniques, including a comparison in terms of the amount of data used to estimate each type of transform. fMLLR transforms estimated on 10s of unsupervised data result in an increase in WER compared with the SI-trained baseline (16.1% vs. 15.0%). When combined with LHUC or SAT-LHUC some of this deterioration was recovered (similar results using LHUC alone were reported in Fig 5.9). For more adaptation data (30s or more) fMLLR improved the accuracies by around 1–2% absolute and combination with LHUC (or SAT-LHUC) resulted in an additional 1% reduction in WER (see also Table 6.6 in the next section for further results).

We also investigated (in a rather unrealistic experiment) how much mismatch in feature space one can normalise in model space with LHUC. To do so, we used a SAT-fMLLR trained model with unadapted PLP features which gave a large increase in WER (26% vs 15%). Then, using unsupervised adaptation targets obtained from the feature-mismatched decoding both LHUC and SAT-LHUC were applied. The results (also presented in Fig. 5.11) indicate that a very large portion of the WER increase can be effectively compensated in model space – more than 8% absolute. As found before, test-only re-parametrisation functions ($\exp(r)$ vs. $2/(1+\exp(-r))$) have negligible impact on the adaptation results, and SAT-LHUC again provides better results.

5.5.6 Adaptation Summary

In this section we summarise our results, applying LHUC and SAT-LHUC to TED, AMI, and Switchboard. Table 5.9 contains results for four IWSLT test sets (dev2010, tst2010, tst2011, and tst2013): in most scenarios SAT-LHUC results in a lower WER than LHUC and both techniques are complementary with SAT-fMLLR training.

Similar conclusions can be drawn from experiments on AMI (Table 6.5) where LHUC and SAT-LHUC were found to effectively adapt ANN and CNN models trained on FBANK features. SAT-LHUC trained ANN models gave the same final results as the more complicated SAT-fMLLR+LHUC system.

On Switchboard, in contrast to other corpora, we observed that test-only LHUC does not match the WERs obtained from SAT-fMLLR models (Table 5.11). The SI system has a WER of 21.7% compared with 20.7% for the test-only LHUC

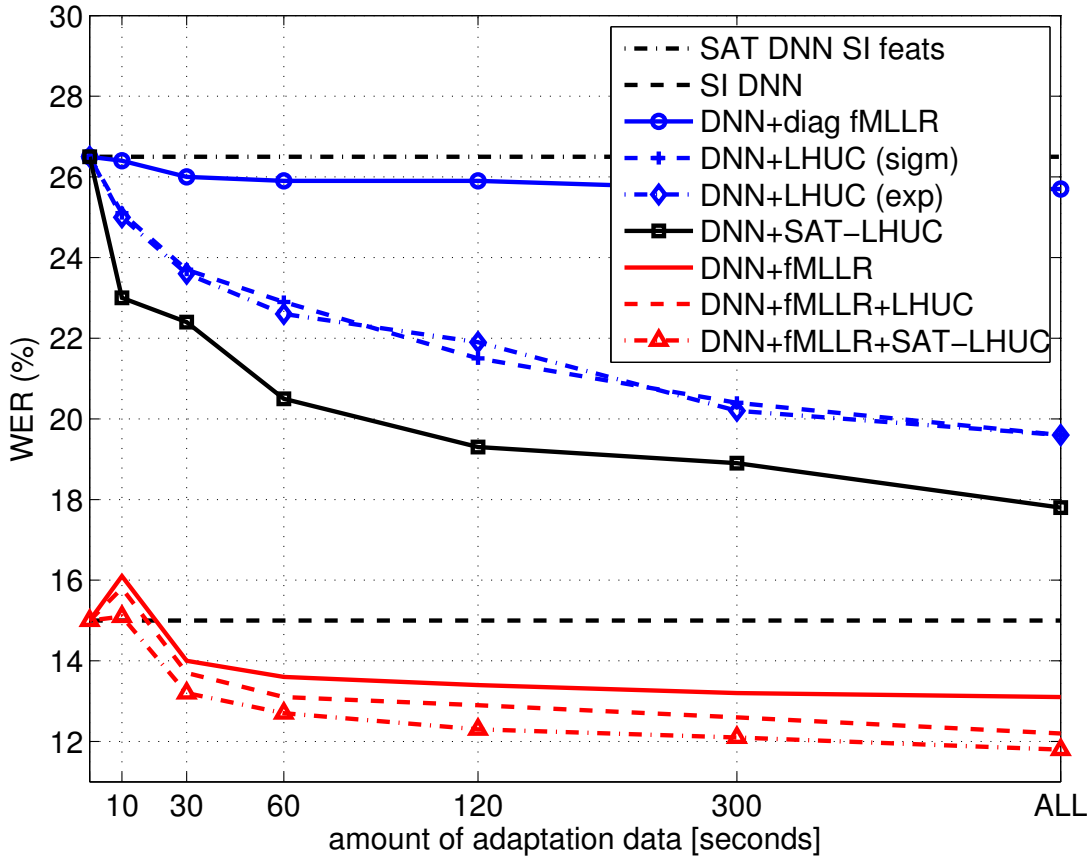


Figure 5.11: WER(%) for LHUC, SAT-LHUC, and SAT-fMLLR (and combinations) on TED tst2010.

Table 5.9: WER (%) on various TED development and test sets from IWSLT12 and IWSLT13 evaluations.

Model	dev2010	tst2010	tst2011	tst2013
ANN	15.4	15.0	12.1	22.1
+LHUC	14.5	12.8	10.9	19.1
+SAT-LHUC	14.0	12.4	10.9	18.0
+fMLLR	14.5	12.9	10.9	20.8
++LHUC	14.1	11.8	10.3	18.4
++SAT-LHUC	13.7	11.6	9.9	17.6

Table 5.10: WER(%) on AMI–IHM

Model	Features	amidev	amieval
ANN	FMLLR	26.2	27.3
+LHUC	FMLLR	25.6	26.2
ANN	FBANK	26.8	29.1
+LHUC	FBANK	25.6	27.1
+SAT-LHUC	FBANK	24.9	26.1

and 20.2% for the SAT-fMLLR system. The improvement obtained using test-only LHUC is comparable to that obtained with other test-only adaptation techniques, e.g. feature-space discriminative linear regression (fDLR) [Seide et al., 2011], but neither of these matches SAT trained feature transform models. This could be due to the fact Switchboard data is narrow-band and as such contains less information for discrimination between speakers [Wester et al., 2015], especially when estimating relevant statistics from small amounts of unsupervised adaptation data. Another potential reason could be related to the fact that the Switchboard part of eval2000 is characterised by a large overlap between training and test speakers – 36 out of 40 test speakers are observed in training [Fiscus et al., 2000], which limits the need for adaptation, but also enables models to learn much more accurate speaker-characteristics during supervised speaker adaptive training.

Adaptation using SAT-LHUC (20.3% WER) almost matches SAT-fMLLR (20.2%). We also observe that LHUC performs relatively better under more mismatched conditions (the Callhome (CHE) subset of eval2000), similar to what we observed

Table 5.11: WER(%) on Switchboard eval2000.

	eval2000		
Model	SWB	CHE	TOTAL
ANN	15.2	28.2	21.7
+LHUC	14.7	26.6	20.7
++SAT-LHUC	14.6	25.9	20.3
+fMLLR	14.2	26.2	20.2
++LHUC	14.2	25.6	19.9
++SAT-LHUC	14.1	25.6	19.9

on TED.

Finally, in Fig 5.12 we show the WERs obtained for 200 speakers across the TED, AMI, and SWBD test sets. We observe that for 89% of speakers LHUC and SAT-LHUC adaptation reduced the WER, and that SAT-LHUC gives a consistent reduction over LHUC.

5.6 LHUC for Factorisation

We applied LHUC to adapt to both the speaker and the acoustic environment. If multi-condition data is available for a speaker, then it is possible to define a set of joint speaker-environment LHUC transforms. Alternatively, we can estimate two set of transforms – for speaker $\mathbf{r}_S$ and for environment $\mathbf{r}_E$ – and then linearly interpolate them, using hyper-parameter α, to derive a combined transform $\hat{\mathbf{r}}_{SE}$ as follows:

$$\xi\left(\hat{\mathbf{r}}^l_{SE}\right) = \alpha\xi\left(\mathbf{r}^l_S\right) + (1-\alpha)\xi\left(\mathbf{r}^l_E\right) \tag{5.10}$$

Notice, that although both types of transforms are estimated in an unsupervised manner we assume that the test environment is known, allowing the correct environmental transform to be selected. This adaptation to the test environment is similar to that of Li et al. [2014a].

We adapted baseline multi-condition trained ANN models [Seltzer et al., 2013] to the speaker ($\mathbf{r}_S$) and the environment ($\mathbf{r}_E$). The $\mathbf{r}_S$ transforms were estimated only on *clean* speech; similarly the environment transforms were estimated for each scenario (one set of $\mathbf{r}_E$ per scenario) using multiple speakers (hence, we have 7 different environmental transforms). To avoid learning joint speaker-environment

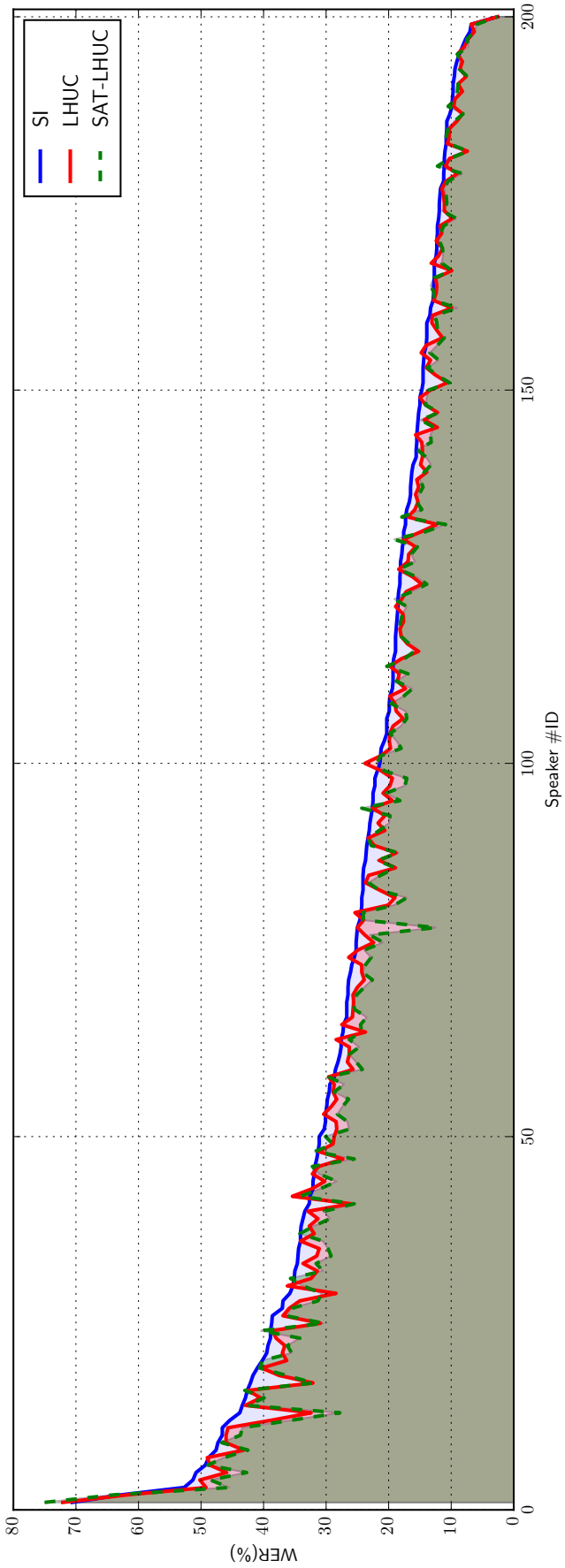


Figure 5.12: Summary of WERs(%) obtained with LHUC and SAT–LHUC adaptation techniques on test speakers of TED, SWBD and AMI corpora (results are sorted in descending WER order for the SI system). For LHUC the average observed improvement per speaker was at 1.6% absolute (7.0% relative). The same statistic for SAT–LHUC was at 2.3% absolute (9.7% relative). The maximum observed WER decrease per speaker was 11.4% absolute (32.7% relative) and 16.0% absolute (50% relative) for LHUC and SAT–LHUC, respectively. WERs decreased for 89% of speakers using LHUC adaptation. LHUC vs. SAT–LHUC results are significant at $p_v<0.001$ level (compare with the significance levels reported in Table 4.8).

Table 5.12: WER(%) results on Aurora 4. Multi-condition ANN model.

Model	A	B	C	D	AVG
ANN	5.1	9.3	9.3	20.8	13.9
ANN + $\mathbf{r}_S$	4.3	9.3	6.9	19.3	13.1
ANN + $\mathbf{r}_E$	5.0	9.0	8.5	19.8	13.3
ANN + $\mathbf{r}_{SE\ JOINT}$	4.5	8.6	7.4	18.3	12.4
ANN + $\hat{\mathbf{r}}_{SE}$, $\alpha = 0.5$	4.6	8.9	7.7	19.1	12.9
ANN + $\hat{\mathbf{r}}_{SE}$, $\alpha = 0.7$	4.5	8.8	7.2	18.9	12.7

transforms the target speaker's data was removed from environment adaptation material (e.g. when estimating transforms for the `restaurant` environment, we use all restaurant data except the one for the target speaker).

The results (Table 5.12) show that both standalone speaker or environment adaptation `LHUC` adaptation improve over an unadapted system (13.1%(S) and 13.3%(E) vs. 13.9%) but, as expected, a single transform estimated jointly on the target speaker and environment has a lower WER (12.4%). However, when interpolated with $\alpha = 0.7$ the result of the factorised model improves to 12.7% WER, although still higher than joint estimation. However, adaptation data for joint speaker-environment adaptation is not available in many scenarios, and the factorised adaptation based on interpolation of distinct transforms estimated separately for speaker and for environment is more flexible. Similar analysis has been carried out for the models trained on clean-only data, as shown in Table 5.13. As expected the clean-trained model benefits more from adaptation to an environment condition rather than to a speaker and joint adaptation offers the best performance. In this scenario, also, factorised transform are better than their standalone usage.

We also trained more competitive models following Rennie et al. [2014]: Maxout [Goodfellow et al., 2013] CNN models were trained using dropout [Srivastava et al., 2014] with an annealing schedule. In this work we used alignments obtained by aligning a corresponding multi-condition model as ground-truth labels, rather than replicating clean alignments to multi-condition data, in contrast to [Rennie et al., 2014]: this is likely to explain differences in the reported baselines (10.9% compared with 10.5% in [Rennie et al., 2014]). The results for the joint optimi-

Table 5.13: WER(%) results on Aurora 4. Clean ANN model.

Model	A	B	C	D	AVG
ANN	4.8	26.2	21.2	43.5	31.7
ANN + $\mathbf{r}_S$	4.7	25.0	15.9	40.6	29.6
ANN + $\mathbf{r}_E$	5.2	19.0	19.6	35.6	27.5
ANN + $\mathbf{r}_{SE\ JOINT}$	4.9	19.7	16.9	35.3	25.1
ANN + $\hat{\mathbf{r}}_{SE}$, $\alpha = 0.5$	4.8	22.0	19.1	38.2	27.5
ANN + $\hat{\mathbf{r}}_{SE}$, $\alpha = 0.3$	4.8	19.1	19.6	36.5	27.0
ANN + $\mathbf{r}_{SE}$ (Oracle)	5.5	17.3	15.1	31.0	22.1

Table 5.14: WER(%) results on Aurora 4. Multi-condition Maxout-CNN model, with and without annealed dropout (AD).

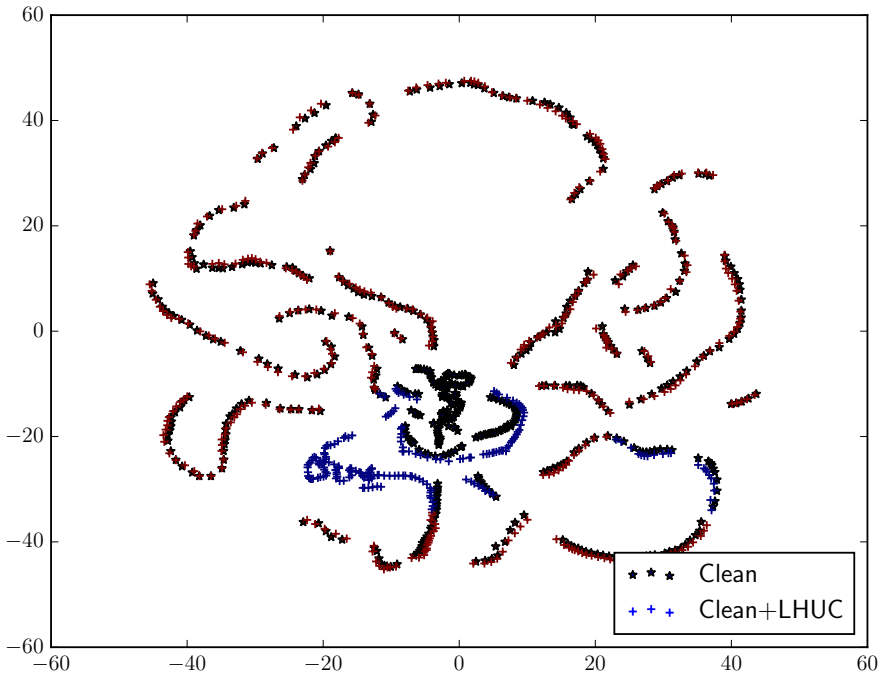
Model	A	B	C	D	AVG
MaxCNN	4.2	7.7	7.9	17.4	11.6
MaxCNN + $\mathbf{r}_{SE\ JOINT}$	3.7	6.3	5.5	14.3	9.5
AD MaxCNN	4.3	7.7	7.2	15.6	10.9
AD MaxCNN + $\mathbf{r}_{SE\ JOINT}$	3.4	5.7	6.1	13.4	8.7

sation are reported in Table 5.14 where one can notice large improvements with unsupervised LHUC adaptation.

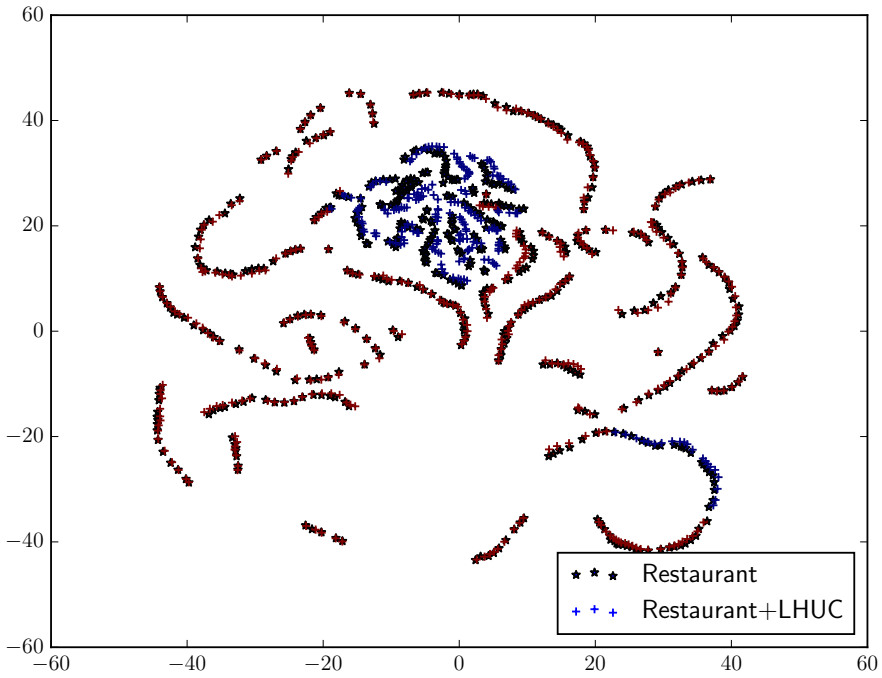
Finally, we visualise the top hidden layer activations of the annealed dropout Maxout CNN using stochastic neighbourhood embedding (tSNE) [van der Maaten and Hinton, 2008] for one utterance recorded under clean and noisy (restaurant) conditions (Fig. 5.13).

5.7 Summary

This chapter introduced the LHUC approach – an unsupervised technique for adaptation of neural network acoustic models in both test-only (LHUC) and SAT (SAT-LHUC) frameworks, evaluating them using four standard speech recognition corpora: TED talks as used in the IWSLT evaluations, AMI, Switchboard, and Aurora4. Our experimental results indicate that both LHUC and SAT-LHUC can



(a)



(b)

Figure 5.13: tSNE plots (best viewed in color) of the top hidden layer before and after adaptation for an utterance recorded in (a) clean and (b) noisy (restaurant) environment, using the annealed dropout maxout CNN. The model can normalise the phonetic space between conditions (brown color), keeping two different spaces for non-speech frames (blue color) under clean and noisy conditions. The effect of `LHUC` is mostly visible for non-speech frames.

provide significant improvements in word error rates (5–23% relative depending on test set and task). `LHUC` adaptation works well unsupervised and with small amounts of data (as little as 10s), is complementary to feature space normalisation transforms such as SAT-fMLLR, and can be used for unsupervised adaptation of sequence-trained ANN acoustic models using a cross-entropy adaptation objective function. Furthermore we have demonstrated that it can be applied in a factorised way, estimating and interpolating separate transforms for adaptation to the acoustic environment and speaker.

Chapter 6

Differentiable Pooling

The content of this chapter is based on [Swietojanski and Renals, 2016], published in IEEE/ACM Transactions on Audio, Speech and Language Processing, which is an extended version of [Swietojanski and Renals, 2015] published at IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).

6.1 Introduction

In this chapter we explore the use of parametrised and differentiable pooling operators for adapting acoustic models. More specifically, we propose to treat pooling operations as speaker-dependent using two types of differentiable parametrisations: L_p-norm pooling and weighted Gaussian pooling (Section 6.2). We show how the pooling parameters may be optimised by maximising the probability of the class given the input data in (Section 6.3), and provide a justification for the use of pooling operators in adaptation in Section 6.4. To evaluate the proposed adaptation approach we performed experiments on three corpora – TED talks, Switchboard conversational telephone speech, and AMI meetings – presenting results on using differentiable pooling for speaker independent acoustic modelling, followed by unsupervised speaker adaptation experiments in which adaptation of the pooling operators is compared (and combined) with learning hidden unit contributions (LHUC) and constrained/feature-space maximum likelihood linear regression (fMLLR).

6.2 Differentiable Pooling

This chapter presents an approach to adaptation by learning hidden layer pooling operators with parameters that can be learned and adapted in a similar way to the other model parameters. The idea of feature pooling originates from Hubel and Wiesel's pioneering study on visual cortex in cats [Hubel and Wiesel, 1962], and was first used in computer vision to combine spatially local features by Fukushima and Miyake [1982]. Pooling in ANNs involves the combination of a set of hidden unit outputs into a summary statistic. Fixed poolings are typically used, such as average pooling (used in the original formulation of convolutional neural networks – CNNs) [LeCun et al., 1989, 1998a] and max pooling (used in the context of feature hierarchies [Riesenhuber and Poggio, 1999] and later applied to CNNs [Ranzato et al., 2007, Boureau et al., 2010]).

Reducing the dimensionality of hidden layers by pooling some subsets of hidden unit activations has become well investigated beyond computer vision, and the max operator has been interpreted as a way to learn piecewise linear activation functions – referred to as Maxout [Goodfellow et al., 2013]. Maxout has been widely investigated for both fully-connected [Miao et al., 2013, Cai et al., 2013, Swietojanski et al., 2014b] and convolutional [Renals and Swietojanski, 2014, Toth, 2014] ANN-based acoustic models. Max pooling, although differentiable, performs a one-from-K selection, and hence does not allow hidden unit outputs to be interpolated, or their combination to be learned within a pool.

There have been a number of approaches to pooling with differentiable operators – *differentiable pooling* – a notion introduced by Zeiler and Fergus [2012] in the context of constructing unsupervised feature extraction for support vector machines in computer vision tasks. There has been some interest in the use of L_p-norm pooling with CNN models [Boureau et al., 2010, Sermanet et al., 2012] in which the sufficient statistic is the p-norm of the group of (spatially-related) hidden unit activations. Fixed order L_p-norm pooling was recently applied within the context of a convolutional neural network acoustic model [Sainath et al., 2013a], where it did not reduce the WER over max-pooling, and as an activation function in a fully-connected ANN [Zhang et al., 2014], where it was found to improve over competitive models, including Maxout and ReLU.

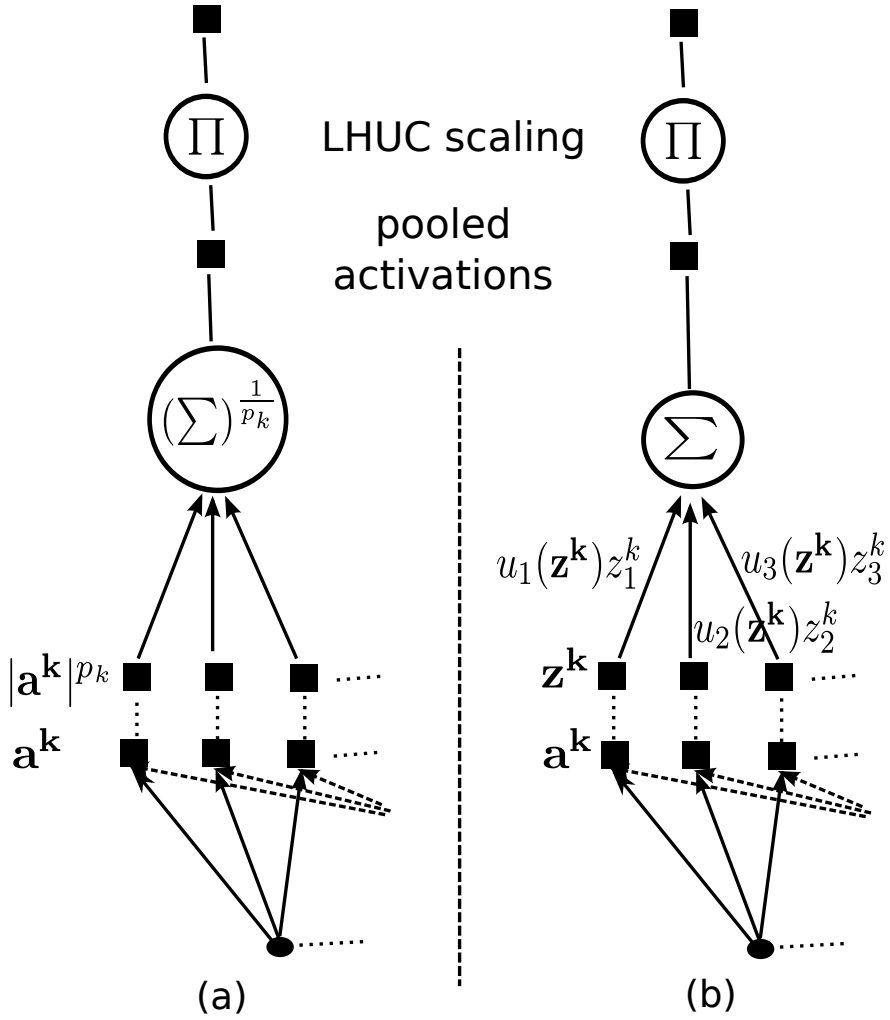


Figure 6.1: Illustration of `Diff-`L_p (a) and `Diff-Gauss` (b) pooling operators. See Sections 6.2.1 and 6.2.2 for further details and explanations of the symbols. `LHUC` scaling follows the method described in Chapter 5 and is used only during adaptation.

6.2.1 L_p-norm (Diff-L_p) pooling

In this approach we pool a set of activations using an L_p-norm. A hidden unit pool is formed by a set R_k of K affine projections which form the input to the kth pooling unit, which we write as an ordered set (vector) $\mathbf{a}^k = \{\mathbf{w}_i^\top \mathbf{x} + b_i\}_{i \in R_k}$, where $\mathbf{x}$ denotes some arbitrary input. The output of the kth pooled unit is produced as an L_p norm:

$$f_{L_p}\left(\mathbf{a}^k; p_k\right) = ||\mathbf{a}^k||_{p_k} = \left(\frac{1}{K}\sum_{i \in R_k} |a_i^k|^{p_k}\right)^{\frac{1}{p_k}} \tag{6.1}$$

where p_k is the learnable norm order for the kth unit, that can be jointly optimised with the other parameters in the model. To ensure that (6.1) satisfies a triangle inequality ($p_k \geq 1$; a necessary property of the norm), during optimisation p_k is re-parametrised as $p_k = \zeta(\rho_k) = \max(1, \rho_k)$, where ρ_k is the actual learned parameter. For the case when $p_k = \infty$ we obtain the max-pooling operator Riesenhuber and Poggio [1999]:

$$||\mathbf{a}^k||_\infty = \max\left(\{|a_i^k|\}_{i \in R_k}\right) \tag{6.2}$$

Similarly, if $p_k = 1$ we obtain absolute average pooling (assuming the pool is normalised by K). We refer to this model as Diff-L_p, and it is parametrised by $\boldsymbol{\theta}_{L_p} = \{\{\mathbf{W}^l, \mathbf{b}^l, \boldsymbol{\rho}^l\}_{l=1}^{L-1}, \mathbf{W}^L, \mathbf{b}^L\}$. Sermanet et al. [2012] investigated fixed-order L_p pooling for image classification, which was applied to speaker independent acoustic modelling by Zhang et al. [2014]. Here we allow each L_p unit in the model to have a learnable order p of Gülçehre et al. [2014], and we use the pooling parameters to perform model-based test-only acoustic adaptation.

6.2.2 Gaussian kernel (Diff-Gauss) pooling

The second pooling approach estimates the pooling coefficients using a Gaussian kernel. We generate the pooling inputs at each layer as:

$$\mathbf{z}^k = \left\{\eta_k \cdot \phi(\mathbf{w}_i^\top \mathbf{x} + b_i)\right\}_{i \in R_k} = \left\{\eta_k \cdot \phi(\mathbf{a}_i^k)\right\}_{i \in R_k} \tag{6.3}$$

where ϕ is a non-linearity (tanh in this work) and $\mathbf{a}^k$ is a set of affine projections as before. A non-linearity is essential as otherwise (contrary to L_p pooling) we would produce a linear transformation through a linear combination of linear projections. η_k is the kth pool amplitude; this parameter is tied and learned

per-pool as this was found to give similar results to per-unit amplitudes (but with fewer parameters), and better results compared to setting to a fixed value $\eta_k = 1.0$ [Swietojanski and Renals, 2015].

Given the activation (6.3), the pooling operation is defined as a weighted average over a set R_k of hidden units, where the k-th pooling unit $f_G(\cdot; \boldsymbol{\theta}^k)$ is expressed as:

$$f_G\left(\mathbf{z}^k; \boldsymbol{\theta}^k\right) = \sum_{i \in R_k} u_i(\mathbf{z}^k; \boldsymbol{\theta}^k) z_i^k \tag{6.4}$$

The pooling contributions $\mathbf{u}(\mathbf{z}^k; \boldsymbol{\theta}^k)$ are normalised to sum to one within each pooling region R_k (6.5) and each weight $u_i(\mathbf{z}^k; \boldsymbol{\theta}^k)$ is coupled with the corresponding value of z_i^k by a Gaussian kernel (6.6) (one per pooling unit) parameterised by the mean and precision, $\boldsymbol{\theta}^k = \{\mu_k, \beta_k\}$:

$$u_i(\mathbf{z}^k; \boldsymbol{\theta}^k) = \frac{v(z_i^k; \boldsymbol{\theta}^k)}{\sum_{i' \in R_k} v(z_{i'}^k; \boldsymbol{\theta}^k)} \tag{6.5}$$

$$v(z_i^k; \boldsymbol{\theta}^k) = \exp\left(-\frac{\beta_k}{2}\left(z_i^k - \mu_k\right)^2\right) \tag{6.6}$$

Similar to L_p-norm pooling, this formulation allows a generalised pooling to be learned – from average ($\beta \to 0$) to max ($\beta \to \infty$) – separately for each pooling unit $f_G(\mathbf{z}^k; \boldsymbol{\theta}^k)$ within a model (see the next section for more details). The `Diff-Gauss` model is thus parametrised by $\boldsymbol{\theta}_G = \{\{\mathbf{W}^l, \mathbf{b}^l, \boldsymbol{\mu}^l, \boldsymbol{\beta}^l, \boldsymbol{\eta}^l\}_{l=1}^{L-1}, \mathbf{W}^L, \mathbf{b}^L\}$.

6.3 Learning Differentiable Poolers

We optimise the acoustic model parameters by minimising the negative log probability of the target HMM tied state given the acoustic observations using gradient descent and error back-propagation [Rumelhart et al., 1986]; the pooling parameters may be updated in a speaker-dependent manner, to adapt the acoustic model to unseen data. In this section we give the necessary partial derivatives for `Diff-`L_p and `Diff-Gauss` pooling.

6.3.1 Learning and adapting `Diff-`L_p pooling

In `Diff-`L_p pooling we learn p_k which we express in terms of ρ, $p_k = \zeta(\rho_k)$. Error back-propagation requires the partial derivative of the pooling region $f_{L_p}(\mathbf{a}^k; \rho_k)$

with respect to ρ_k, which is given as:

$$\frac{\partial f_{L_p}(\mathbf{a}^k;\rho_k)}{\partial \rho_k} = \left(\frac{\sum_{i\in R_k} \log(|a_i^k|)\cdot |a_i^k|^{p_k}}{p_k \sum_{i\in R_k} |a_i^k|^{p_k}} - \frac{\log \sum_{i\in R_k} |a_i^k|^{p_k}}{p_k^2} \right) \frac{\partial \zeta(\rho_k)}{\partial \rho_k} f_{L_p}(\mathbf{a}^k;\rho_k) \tag{6.7}$$

where $\partial\zeta(\rho_k)/\partial\rho_k = 1$ when $p_k > 1$ and 0 otherwise. The back-propagation through the norm itself is implemented as:

$$\frac{\partial f_{L_p}(\mathbf{a}^k;p_k)}{\partial \mathbf{a}^k} = \frac{\mathbf{a}^k \circ |\mathbf{a}^k|^{p_k-2}}{\sum_{i\in R_k} |a_i^k|^{p_k}} \circ \mathbf{G}^k \tag{6.8}$$

where $\circ$ represents the element-wise Hadamard product, and $\mathbf{G}^k$ is a vector of $f_{L_p}(\mathbf{a}^k;p_k)$ activations repeated K times, so the resulting operation can be fully vectorised:

$$\mathbf{G}^k = \left[f_{L_p}(\mathbf{a}^k;p_k)^1, \ldots, f_{L_p}(\mathbf{a}^k;p_k)^K \right]^\top \tag{6.9}$$

Normalisation by K in (6.1) is optional (see also Section 6.5.1) and the partial derivatives in (6.7) and (6.8) hold for the un-normalised case also: the effect of this is taken into account in the forward activation $f_{L_p}(\mathbf{a}^k;p_k)$.

Since (6.7) and (6.8) are not continuous everywhere, they need to be stabilised when $\sum_{i\in R_k} |a_i^k|^{p_k} = 0$. When computing (6.7) it is also necessary to ensure that $a_i^k > 0$. In practise, we threshold each element to have at least a value $\epsilon = 10^{-8}$ if $a_i^k < \epsilon$. Note, this stabilisation concerns L_p units only.

6.3.2 Learning and adapting `Diff-Gauss` pooling regions

To learn the `Diff-Gauss` pooling parameters, we require the partial derivatives $\partial f_G(\mathbf{z}^k)/\partial\mu_k$ and $\partial f_G(\mathbf{z}^k)/\partial\beta_k$ to update pooling parameters, as well as $\partial f_G(\mathbf{z}^k)/\partial\mathbf{z}^k$ in order to back-propagate error signals to lower layers.

One can compute the partial derivative of (6.4) with respect to the input activations $\mathbf{z}^k$ as:

$$\frac{\partial f_G(\mathbf{z}^k)}{\partial \mathbf{z}^k} = \left[(\mathbf{z}^k)^\top \left(\mathbf{J}_\mathbf{u}(\mathbf{v}(\mathbf{z}^k)) \mathbf{J}_\mathbf{v}(\mathbf{z}^k) \right) + \mathbf{u}(\mathbf{z}^k)^\top \right]^\top \tag{6.10}$$

where $\mathbf{J}_\mathbf{u}(\mathbf{v}(\mathbf{z}^k))$ is the Jacobian representing the partial derivative $\partial u(\mathbf{z}^k)/\partial v(\mathbf{z}^k)$:

$$\mathbf{J}_\mathbf{u}(\mathbf{v}(\mathbf{z}^k)) = \frac{\partial \mathbf{u}(\mathbf{z}^k)}{\partial \mathbf{v}(\mathbf{z}^k)} = \begin{bmatrix} \frac{\partial u(z_1^k)}{\partial v(z_1^k)} & \cdots & \frac{\partial u(z_1^k)}{\partial v(z_K^k)} \\ \vdots & \ddots & \vdots \\ \frac{\partial u(z_K^k)}{\partial v(z_1^k)} & \cdots & \frac{\partial u(z_K^k)}{\partial v(z_K^k)} \end{bmatrix} \tag{6.11}$$

whose elements can be computed as:

$$\frac{\partial u(\mathbf{z}^k)}{\partial v(z_i^k)} = \Big(\sum_{m \in R_k} v(z_m^k) \Big)^{-1} \Big(1 - u(z_i^k) \Big) \tag{6.12}$$

$$\frac{\partial u(\mathbf{z}^k)}{\partial v(z_{i'}^k)} = \Big(\sum_{m \in R_k} v(z_m^k) \Big)^{-1} \Big(-u(z_i^k) \Big) \tag{6.13}$$

Likewise, $\mathbf{J}_{\mathbf{v}}(\mathbf{z}^k)$ represents the Jacobian of the kernel function $v(\mathbf{z}^k)$ in (6.6) with respect to $\mathbf{z}^k$:

$$\mathbf{J}_{\mathbf{v}}(\mathbf{z}^k) = \frac{\partial \mathbf{v}(\mathbf{z}^k)}{\partial \mathbf{z}^k} = \begin{bmatrix} \frac{\partial v(z_1^k)}{\partial z_1^k} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{\partial v(z_K^k)}{\partial z_K^k} \end{bmatrix} \tag{6.14}$$

and the elements of $\mathbf{J}_{\mathbf{v}}(\mathbf{z}^k)$ can be computed as:

$$\frac{\partial v(z_i^k)}{\partial z_i^k} = -\beta_k (z_i^k - \mu_k) v(z_i^k) \tag{6.15}$$

Similarly, one can obtain the gradients with respect to the pooling parameters $\boldsymbol{\theta}^k$. In particular, for β_k, the gradient is:

$$\frac{\partial f_G(\mathbf{z}^k)}{\partial \beta_k} = \sum_{i \in R_k} \Big[(\mathbf{z}^k)^\top \Big(\mathbf{J}_{\mathbf{u}}(\mathbf{v}(\mathbf{z}^k)) \mathbf{J}_{\mathbf{v}}(\beta_k) \Big) \Big]_i \tag{6.16}$$

where $\mathbf{J}_{\mathbf{v}}(\beta_k) = \partial \mathbf{v}(\mathbf{z}^k) / \partial \beta_k$ and $\partial v(z_i^k) / \partial \beta_k$ is:

$$\frac{\partial v(z_i^k)}{\partial \beta_k} = -\frac{1}{2} \Big(z_i^k - \mu_k \Big)^2 v(z_i^k) \tag{6.17}$$

The corresponding gradient for $\partial f_G(\mathbf{z}^k) / \partial \mu_k$ is obtained below (6.18). Notice, that $\partial v(z_i^k) / \partial z_i^k$ (6.15) and $\partial v(z_i^k) / \partial \mu_k$ (6.19) are symmetric, hence $\mathbf{J}_{\mathbf{v}}(\mu_k) = -\mathbf{J}_{\mathbf{v}}(\mathbf{z}^k)$, and to compute $\partial f_G(\mathbf{z}^k) / \partial \mu_k$ one can reuse the $(\mathbf{z}^k)^\top \mathbf{J}_{\mathbf{u}}(\mathbf{v}(\mathbf{z}^k)) \mathbf{J}_{\mathbf{v}}(\mathbf{z}^k)$ term in (6.10), as follows:

$$\begin{aligned} \frac{\partial f_G(\mathbf{z}^k)}{\partial \mu_k} &= \sum_{i \in R_k} \Big[(\mathbf{z}^k)^\top \Big(\mathbf{J}_{\mathbf{u}}(\mathbf{v}(\mathbf{z}^k)) \mathbf{J}_{\mathbf{v}}(\mu_k) \Big) \Big]_i \\ &= - \sum_{i \in R_k} \Big[(\mathbf{z}^k)^\top \Big(\mathbf{J}_{\mathbf{u}}(\mathbf{v}(\mathbf{z}^k)) \mathbf{J}_{\mathbf{v}}(\mathbf{z}^k) \Big) \Big]_i \end{aligned} \tag{6.18}$$

$$\frac{\partial v(z_i^k)}{\partial \mu_k} = -\frac{\partial v(z_i^k)}{\partial z_i^k} = \beta_k (z_i^k - \mu_k) v(z_i^k) \tag{6.19}$$

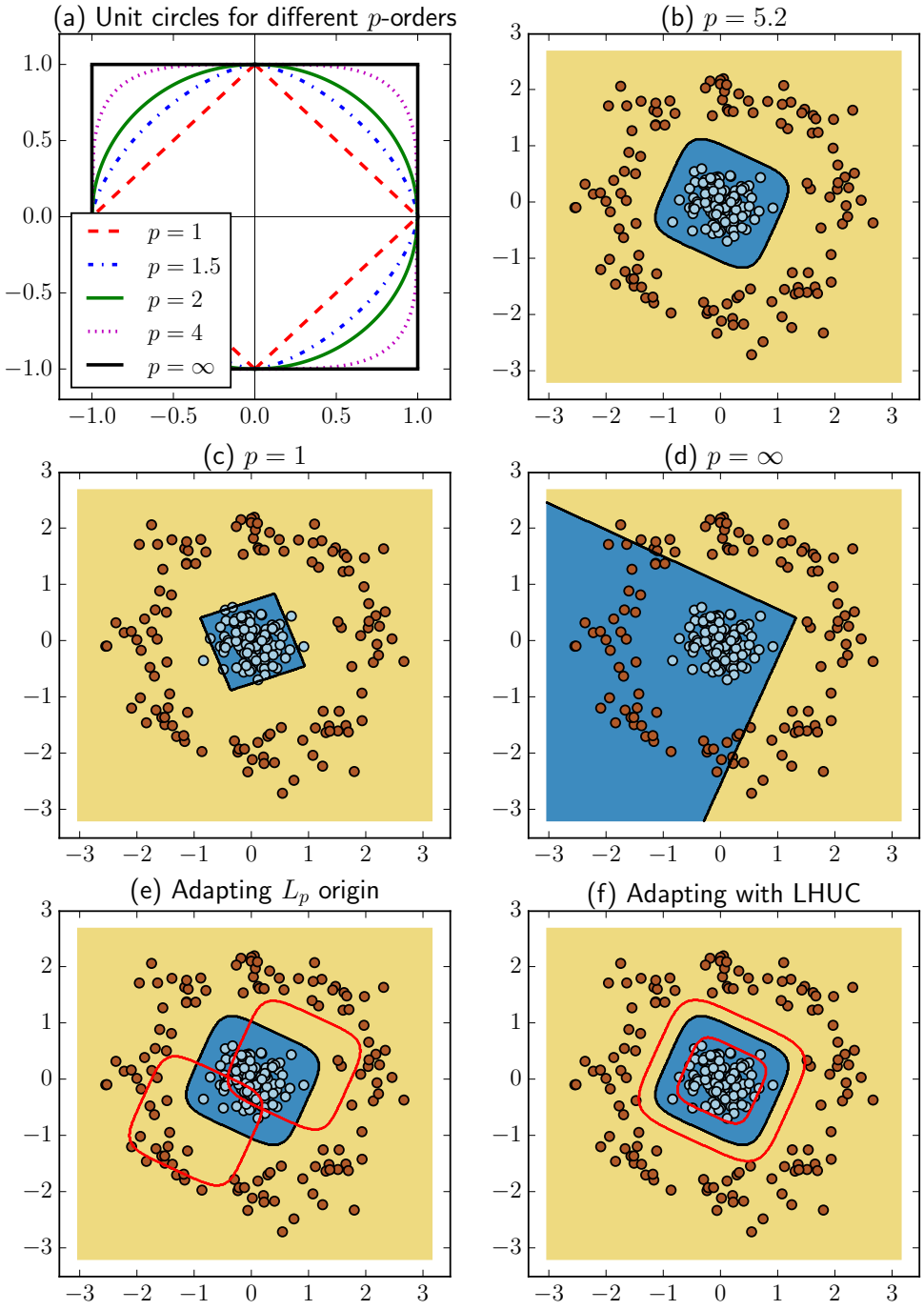


Figure 6.2: Illustration of representational efficiency and adaptation principles behind L_p unit. (a) unit circles as obtained under different norm p-orders. (b) An example decision boundary for some two-class toy data. The `Diff-`L_p model is built out of one L_p unit (with $K = 2$ linear inputs) and is able to draw highly non-linear decision-regions. (c) The model from (b) with $p = 1.0$ and (d) $p = \infty$. Red contours of the bottom two plots illustrate (e) the effect of adaptation of the origin (biases) of the linear inputs $\mathbf{a}^k$ and (f) the effect of `LHUC` scaling. Further description in Section 6.4. (Best viewed in colour.)

6.4 Representational efficiency of pooling units

The aim of model-based ANN adaptation is to alter the learned speaker independent representation in order to improve the classification accuracy for data from a possibly mismatched test distribution. Owing to the highly distributed representations that are characteristic of ANNs, it is rarely clear which parameters should be adapted in order generalise well to a new speaker or acoustic condition.

Pooling enables decision boundaries to be altered, through the selection of relevant hidden features, while keeping the parameters of the feature extractors (the hidden units) fixed: this is similar to `LHUC` adaptation Swietojanski et al. [2016]. The pooling operators allow for a geometrical interpretation of the decision boundaries and how they will be affected by a constrained adaptation – the units within the pool are jointly optimised given the pooling parametrisation, and share some underlying relationship within the pool.

This is visualised for L_p units in Fig. 6.2. Fig. 6.2 (a) illustrates the unit circles obtained by solving $||\mathbf{a}^k||_p = d$ for different orders p, with $d = 1.0$ and a pool of $K = 2$ linear inputs $\mathbf{a}^k$. Such an L_p unit is capable of closed-region decision boundaries, illustrated in Fig. 6.2 (b). The distance threshold d is implicitly learned from data (through the $\mathbf{a}^k$ parameters given p), resulting in an efficient representation Gülçehre et al. [2014], Zhang et al. [2014] compared with representing such boundaries using sigmoid units or ReLUs, which would require more parameters. Figs. 6.2 (c) and (d) show how those boundaries are affected when $p = 1$ (average pooling) and $p = \infty$ (max pooling), while keeping $\mathbf{a}^k$ fixed. As shown in Section 6.5 we found that updating p is an efficient and relatively low-dimensional way to adjust decision boundaries such that the the model's accuracy on the adaptation data distribution improves.

It is also possible to update the biases (Fig. 6.2 (e), red contours) and the `LHUC` amplitudes (Fig. 6.2 (f), red contours). We experimentally investigate how each approach impacts adaptation WER in Section 6.5.2. Although models implementing `Diff-Gauss` units are theoretically less efficient in terms of SI representations compared to L_p units, and comparable to standard fully-connected models, the pooling mechanism still allows for more efficient (in terms of number of SD parameters) speaker adaptation.

6.5 Results

6.5.1 Baseline speaker independent models

The structures of the differentiable pooling models were selected such that the number of parameters was comparable to the corresponding baseline DNN models. For the `Diff-`L_p and `Diff-`L_2 types, the resulting models utilised non-overlapping pooling regions of size $K = 5$, with 900 L_p-norm units per layer. The `Diff-Gauss` models had pool sizes set to $K = 3$ (this was found to work best in our previous work [Swietojanski and Renals, 2015]) which (assuming non-overlapping regions) results in 1175 pooling units per layer. Those parameters were optimised on our TED development set `dev2010`.

6.5.1.1 Training speaker independent `Diff-`L_2 and `Diff-`L_p models

For both `Diff-`L_p and `Diff-`L_2 we trained with an initial learning rate of .008 (for MFCC, PLP, FBANK features) and .006 (for fMLLR features). The learning rate was adjusted using the `newbob` learning scheme [Renals et al., 1992] based on the validation frame error rate. We found that applying explicit pool normalisation in (6.1) gave consistently higher error rates (typically an absolute increase of 0.3% WER): hence we used un-normalised L_p units in all experiments. We did not apply post-layer normalisation [Zhang et al., 2014]. Instead, after each update we scaled the columns (i.e. each a_i^k) of the fully connected weight matrices such that their L_2 norms were below a given threshold (set to 1.0 in this work). For `Diff-`L_p models we initialised $p = 2.0$. Those parameters were optimised on TED and directly applied without further tuning for the other two corpora. In this work we have focussed on adaptation; Zhang et al. [2014] have reported further speaker independent experiments for fixed order L_p units.

6.5.1.2 Training speaker independent `Diff-Gauss` models

The initial learning rate was set to 0.08 (regardless of the feature type), again adjusted using `newbob`. Initial pooling parameters were sampled randomly from normal distribution: $\mu \sim \mathcal{N}(0, 1)$ and $\beta \sim \mathcal{N}(1, 0.5)$. Otherwise, the hyper-parameters were the same as for the baseline ANN models.

Table 6.1: Baseline WER(%) SI results on selected test sets of our benchmark corpora.

Model	TED `tst2010`	AMI `amieval`	SWBD `eval2000`
ANN	15.0	29.1	22.1
`Diff-Gauss`	14.6	29.0	21.4
`Diff-`L_2	14.6	28.5	21.3
`Diff-`L_p	14.5	27.6	21.3

6.5.1.3 Baseline speaker independent results

Table 6.1 gives speaker independent results for each of the considered model types. The `Diff-Gauss` and `Diff-`L_2/`Diff-`L_p models have comparable WERs, with a small preference towards `Diff-`L_p in terms of the final WER on TED and AMI; all have lower average WER than the baseline ANN. The gap between the pooled models increases on AMI data where `Diff-`L_p has a substantially lower WER (3.2% relative) than the fixed order `Diff-`L_2 which in turn has a lower WER than the other two models (`Diff-Gauss` and baseline ANN) by 2.1% relative.

Fig. 6.3 gives more insight into the `Diff-`L_p models by showing how the final distributions of the learned order p differ across AMI, TED and SWBD corpora. p deviates more from its initialisation in the lower layers of the model; there is also a difference across corpora. This follows the intuition of how a multi-layer network builds its representation: lower layers are more dependent on acoustic variabilities, normalising for such effects, and hence feature extractors may differ across datasets – in contrast to the upper layers which rely on features abstracted away from the acoustic data. For these corpora, the order p rarely exceeded 3, sometimes dropping below 2 – especially for layer 1 with SWBD data. However, most L_p units, especially in higher layers, tend to have $p \sim 2$. This corresponds to previous work [Zhang et al., 2014] in which fixed $L_{p=2}$ units tended to obtain lower WER. A similar analysis of `Diff-Gauss` pooling does not show large data-dependent differences in the learned pooling parameters.

Training speed: Table 6.2 shows the average training speeds for each of the considered models. Training pooled units is significantly more expensive than training baseline ANN models. This is to be expected as the pooling operations cannot be easily and fully vectorised. In our implementation training the `Diff-Gauss` or `Diff-`L_p models is about 40% slower than training a baseline

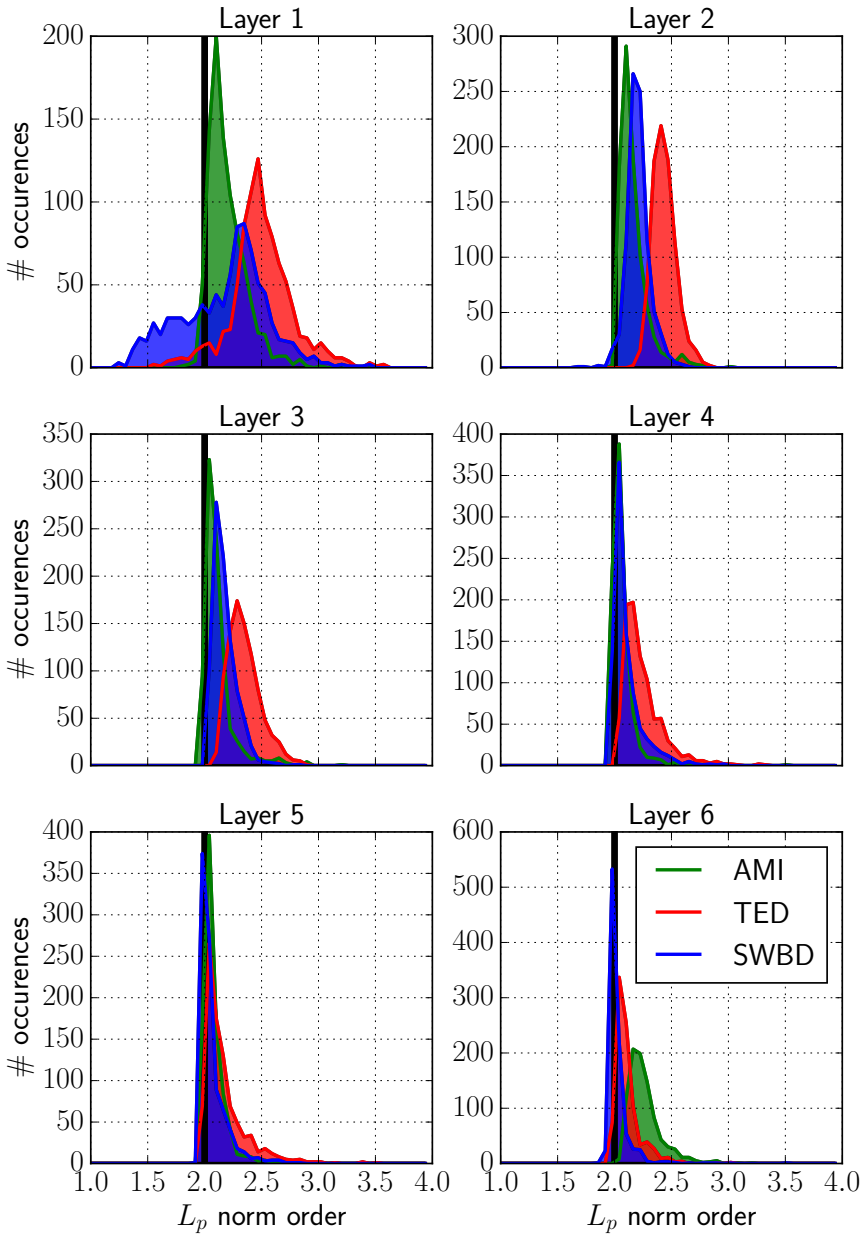


Figure 6.3: L_p orders for the three corpora used in this work. Particular models share the same structure of hidden layers (the same number of L_p units per layer – 900), though both dimensionality of the output layers as well as the acoustic features used to train each model, are different. Vertical black line at 2 denotes an initial p setting of L_p units.

Table 6.2: Average training speeds [frames/second] as obtained for each model type on SWBD data and GTX980 GPGPU boards.

ANN	`Diff-Gauss`	`Diff-`L_2	`Diff-`L_p
9k	5.2k	7.1k	5.4k

Table 6.3: WER(%) results for different subsets of adapted parameters on TED (`tst2010`), AMI(`amieval`) and SWBD (`eval2000`) test-sets. L - #layers, P - #pooling units in layer, K - pool size

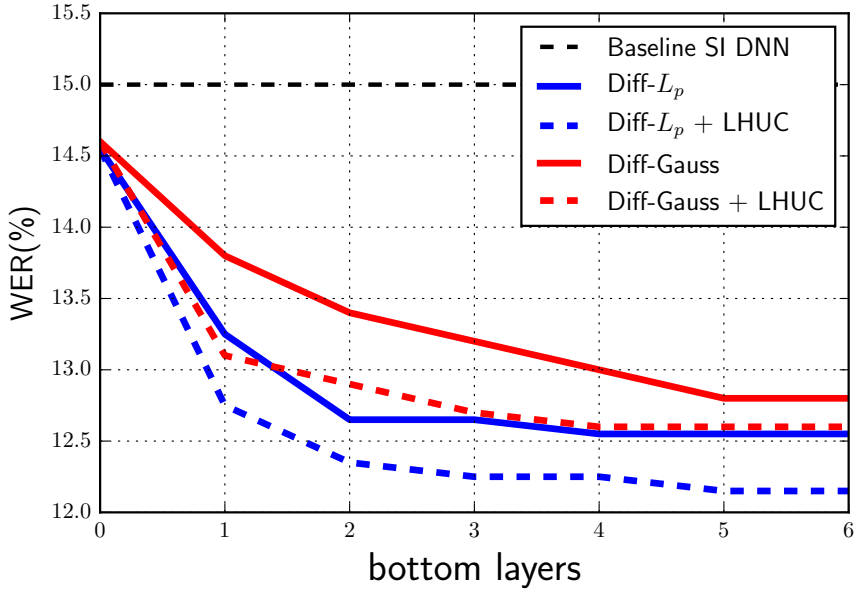
Model	#SD Parameters	TED	AMI	SWBD
`Diff-`L_p	-	14.5	27.6	21.3
+ `LHUC`	$P(L-1)$	12.8	25.8	20.5
+ Update p	$P(L-1)$	12.5	25.8	20.1
++ Update b	$(P+PK)(L-1)$	12.3	25.5	20.5
++ `LHUC`	$2P(L-1)$	12.3	25.6	20.0

ANN. Not optimising p during training (6.7) decreases the gap to about 20% slower. This indicates that training using fixed L_2 units, and then adapting the order p in a speaker adaptive manner could make a good compromise.

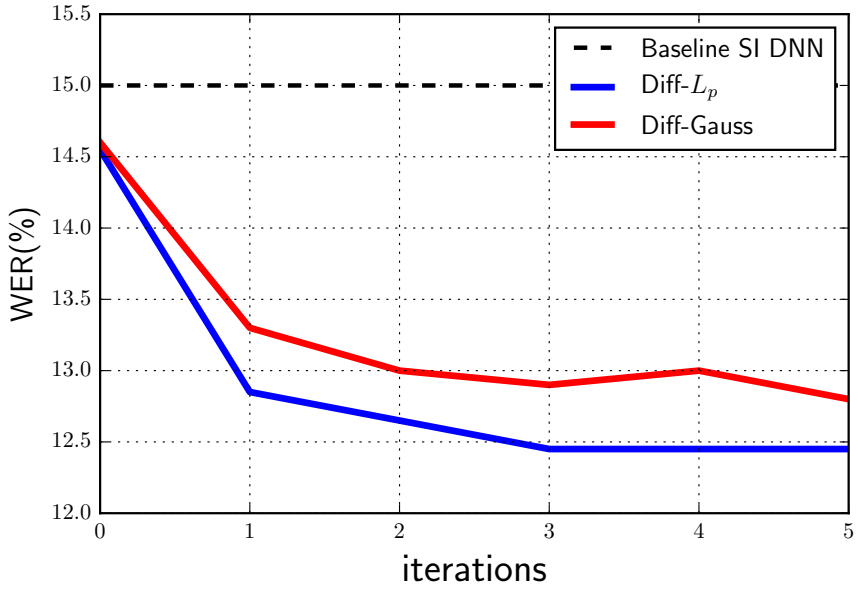
6.5.2 Adaptation experiments

We initially used the TED talks corpus to investigate how WERs are affected by adapting different layers in the model. The results indicated that adapting pooling operators in the bottom layer brings the largest drop in WER; however, adapting more layers in the same way further improves the accuracy for both `Diff-`L_p and `Diff-Gauss` models (Fig. 6.4 (a)). Since obtaining the gradients for the pooling parameters at each layer is inexpensive compared to the overall back-propagation, and adapting bottom layer gives largest gains, in the remainder of this work we adapt all pooling units. Similar trends hold when pooling adaptation is combined with `LHUC` adaptation, which on `tst2010` improves the accuracies by 0.2-0.3% absolute.

Fig. 6.4 (b) shows WER vs. the number of adaptation iterations. The results indicate that one adaptation iteration is sufficient and, more importantly, the model does not overfit when more iterations are used. This suggests that it is not necessary to regularise the model carefully (by Kullback-Leibler divergence [Yu



(a)



(b)

Figure 6.4: WER(%) on `tst2010` as a function of a) number of bottom layers adapted with pooling operators and (optional) `LHUC` transforms and b) number of adaptation iterations

Table 6.4: WER(%) results for different subsets of adapted parameters of `Diff-Gauss` model on TED (`tst2010`), AMI(`amieval`) and SWBD (`eval2000`) test-sets. L - #layers, P - #pooling units in layer

Model	#SD Parameters	TED	AMI	SWBD
`Diff-Gauss`	-	14.6	29.0	21.4
+ `LHUC`	$P(L-1)$	12.8	-	-
+ Update μ	$P(L-1)$	13.1	-	-
+ Update β	$P(L-1)$	13.1	-	-
+ Update η	$P(L-1)$	12.7	-	-
+ Update μ, β	$2P(L-1)$	12.8	27.3	20.7
++ `LHUC`	$3P(L-1)$	12.5	27.0	20.4
++ Update η	$3P(L-1)$	12.3	26.9	20.3

et al., 2013b], for instance) which is usually required when weights that directly transform the data are adapted. In the remainder, we adapt all models with a learning rate of 0.8 for three iterations (optimised on `dev2010`).

Table 6.3 shows the effect of adapting different pooling parameters (including `LHUC` amplitudes) for L_p units. Updating only p, rather than any other stand-alone pooling parameter, gives a lower WER than `LHUC` adaptation with the same number of parameters (cf Fig. 6.2); however, updating both brings further reductions in WER. Adapting the bias is more data-dependent with a substantial increase in WER for SWBD; this also significantly increases the number of adapted parameters. Hence we adapted either p alone, or p with `LHUC` in the remaining experiments

Table 6.4 shows similar analysis but for `Diff-Gauss` model. For `Diff-Gauss`, it is beneficial to update both $\boldsymbol{\mu}$ and $\boldsymbol{\beta}$ (as in Swietojanski and Renals [2015]), and `LHUC` was also found to be complementary. Notice, adapting with `LHUC` scalers is similar to altering η in eq. (6.3) (assuming η is tied per pool, as mentioned in Section 6.2.2). As such, new parameters need not be introduced to adapt `Diff-Gauss` with `LHUC` as it is the case for `Diff-`L_p units. In fact, last two rows of Table 6.4 show that jointly updating μ, β and η gives lower WER than updating μ, β and applying `LHUC` after pooling (see Fig. 6.1).

Analysis of `Diff-`L_p: Fig. 6.5 shows how the distribution of p changes after the `Diff-`L_p model adapts to each of the 28 speakers of `tst2013`. We plot the speaker independent histograms as well as the contours of the mean

bin frequencies for each layer. For the adapted models the distributions of p become less dispersed, especially in higher layers, which can be interpreted as shrinking the decision regions of particular L_p units (cf Fig. 6.2). This follows the intuition that speaker adaptation involves reducing the variability that needs to be modelled, in contrast to the speaker independent model.

Taking into account the increased training time of `Diff-`L_p models, one can also consider training fixed order `Diff-`L_2 [Zhang et al., 2014], adapting p using (6.7). The results in Fig. 6.5, as well as later results, cover this scenario. The adapted `Diff-`L_2 models display a similar trend in the distribution of p to the `Diff-`L_p models.

Analysis of `Diff-Gauss`: We performed a similar investigation on the learned `Diff-Gauss` pooling parameters (Fig. 6.6). In the bottom layers they are characterised by a large negative means and positive precisions which has the effect of turning off many units. After adaptation, some of them become more active, which can be seen based on shifted distributions of adapted pooling parameters in Fig. 6.6. The adaptation with `Diff-Gauss` has a similar effect as to the adaptation of slopes and amplitudes [Zhao et al., 2015, Zhang and Woodland, 2015], but adapts K times fewer parameters.

6.5.2.1 Amount of adaptation data

We investigated the effect of the amount of adaptation data by randomly selecting adaptation utterances from `tst2010` to give totals of 10s, 30s, 60s, 120s, 300s and more speaker-specific adaptation data per talker (Fig. 6.7 (a)). The WERs are an average over three independent runs, each sampling a different set of adaptation utterances (we did more passes in the previous chapter and [Swietojanski and Renals, 2015], however, both `LHUC` and differentiable pooling operators were not very sensitive to this aspect, resulting in small error bars between different results obtained with different random utterances). The `Diff-`L_p models offer lower WER and more rapid adaptation, with 10s of adaptation data resulting in a decrease in WER by 0.6% absolute (3.6% relative) which further increases up to 2.1% absolute (14.4% relative) when using all the speaker's data in an unsupervised manner. `Diff-Gauss` is comparable in terms of WER to a ANN adapted with `LHUC`. In addition, both methods are complementary to `LHUC` adaptation, as well as to feature-space adaptation with fMLLR (Tables 6.6 and 6.7).

In order to demonstrate the modelling capacities of the different model-based

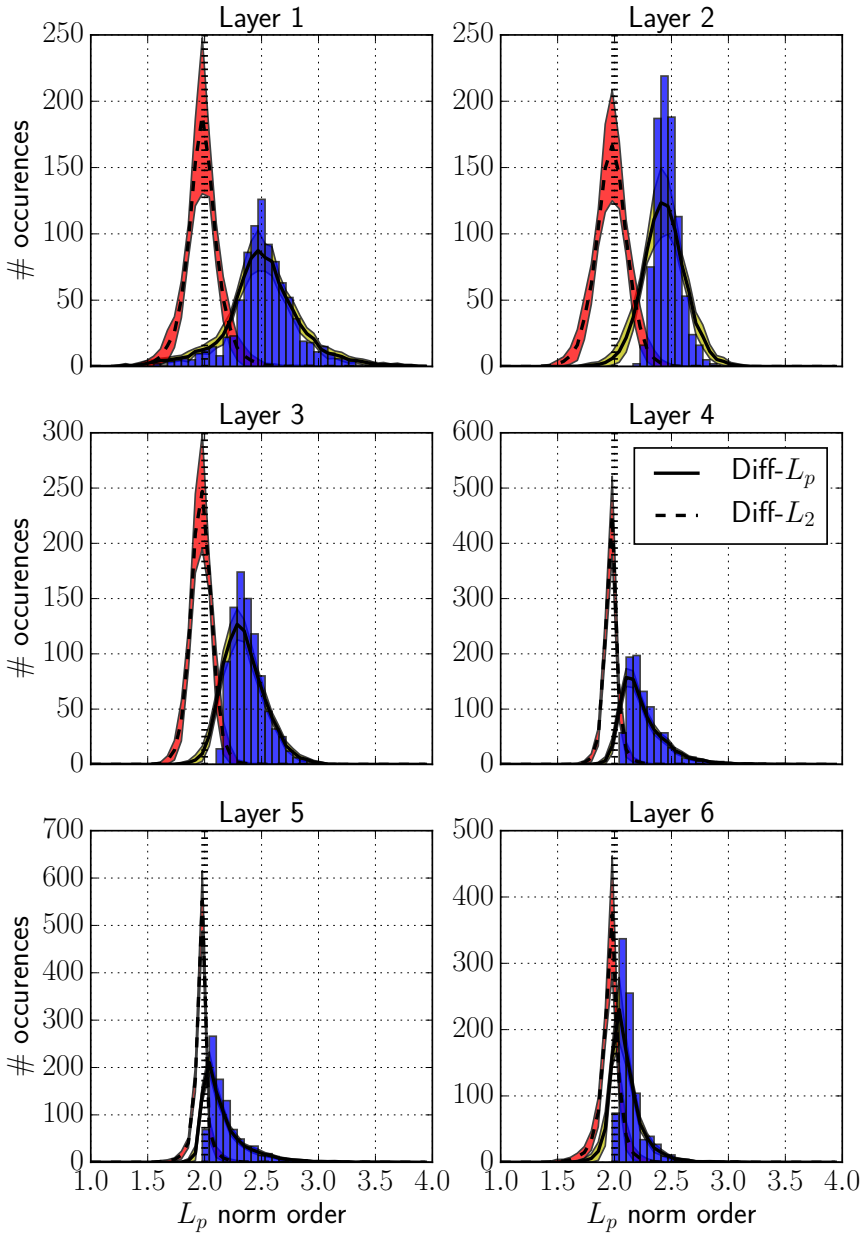


Figure 6.5: Layer-wise histograms of learned L_p orders (speaker independent) on TED data (blue). The vertical line (dashed-black) at 2 is the initial value of p; the black solid line denotes the mean contour ($\pm$ standard deviations in yellow) of the distribution of p obtained after adaptation to 28 speakers of `tst2013`. Likewise, the dashed black line is the mean of the adapted L_p orders ($\pm$ standard deviations in red) starting from a fixed-order `Diff-`L_2 speaker independent model. (Best viewed in colour.)

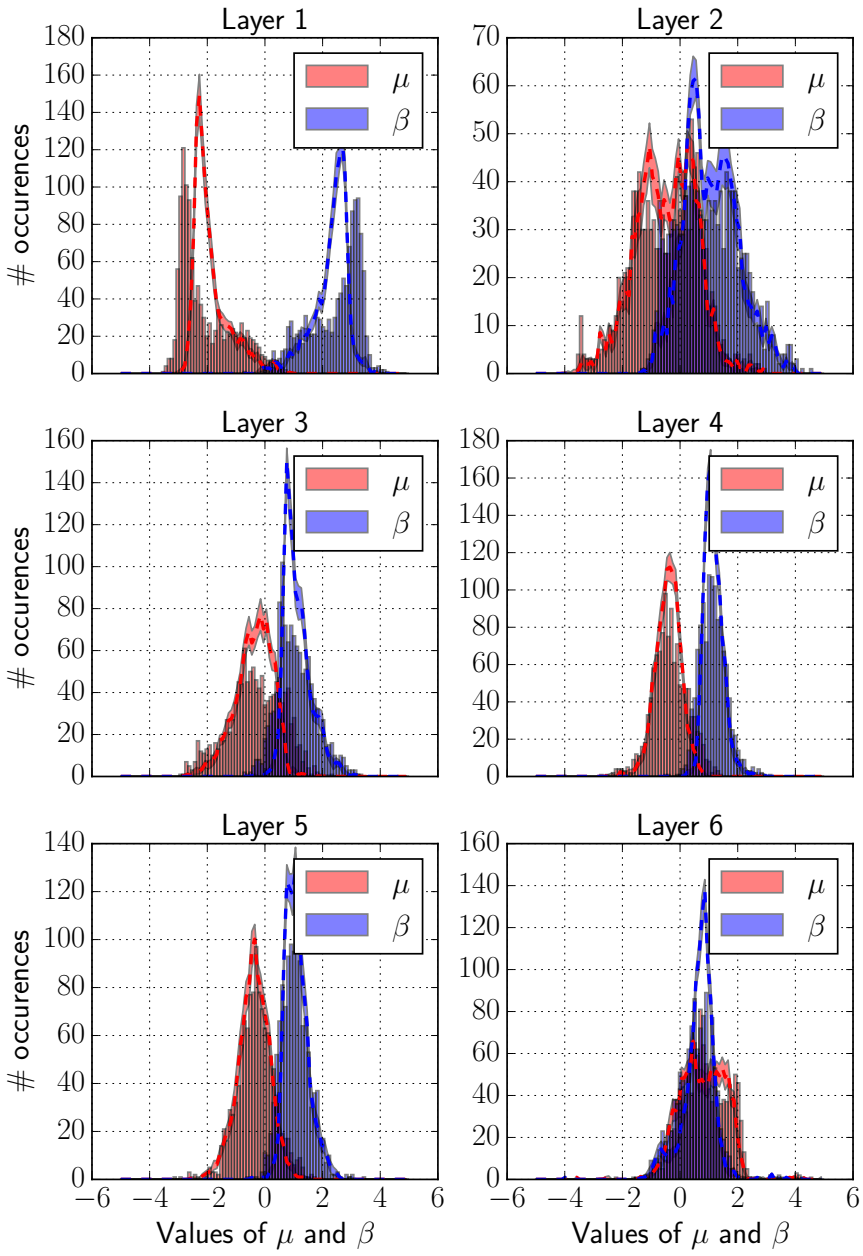
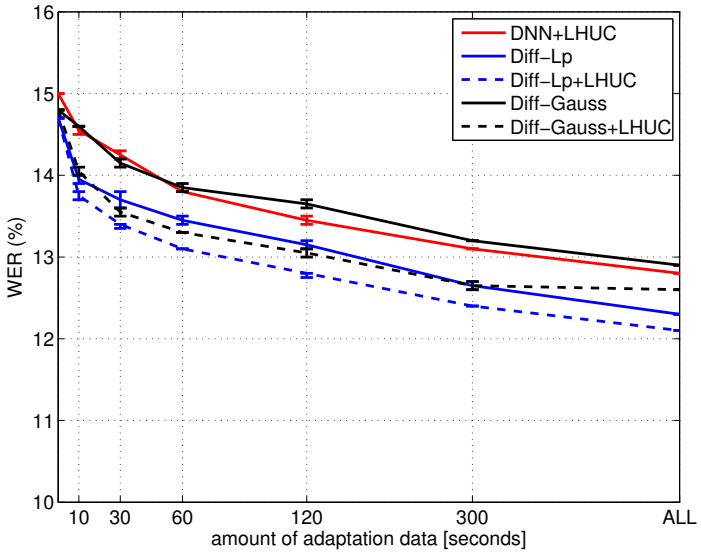
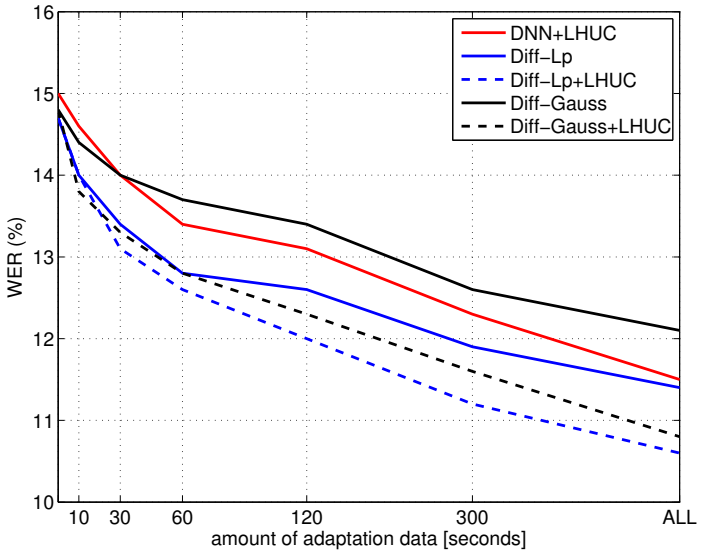


Figure 6.6: Layer-wise histograms of learned `Diff-Gauss` pooling parameters $\{\mu, \beta\}$ during speaker independent training on TED. We also plot the altered mean contours ($\pm$ standard deviation) of the adapted pooling parameters on 28 speakers of `tst2013`. (Best viewed in color.)



(a)



(b)

Figure 6.7: WER(%) on `tst2010` for different amounts of adaptation data with (a) unsupervised and (b) oracle adaptation targets. (c) Adaptation accuracies on `Diff-Gauss` as the quality of adaptation targets differs owing to the language model used to re-score the first-pass hypothesis.

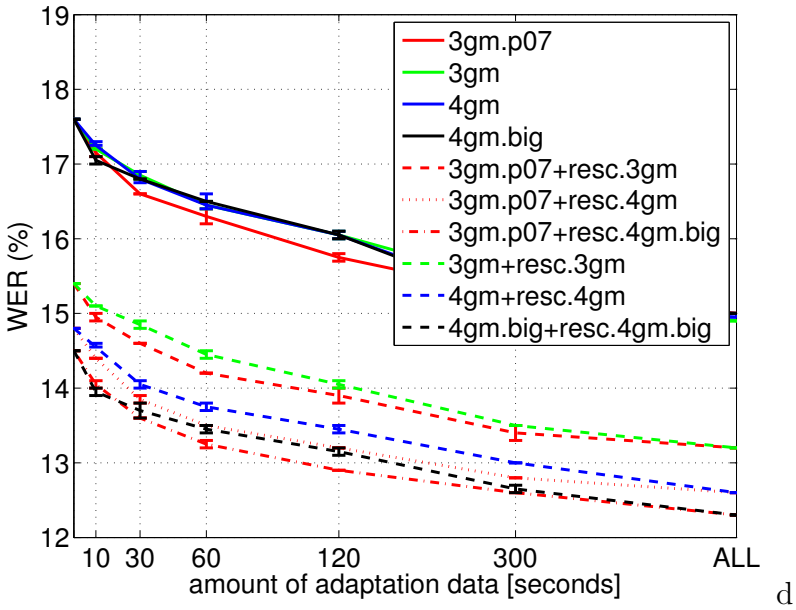
adaptation techniques, we carried out a supervised adaptation (oracle) experiment in which the adaptation targets were obtained by aligning the audio data with reference transcripts (Fig. 6.7 (b)). We do not refine what the model knows about speech, nor the way it classifies it (the feature receptors and output layer are fixed during adaptation and remain speaker independent), but show that the re-composition and interpolation of these basis functions to approximate the unseen distribution of adaptation data is able to decrease the WER by 26.7% relative for `Diff-`L_p + `LHUC` scenario.

6.5.2.2 Quality of adaptation targets

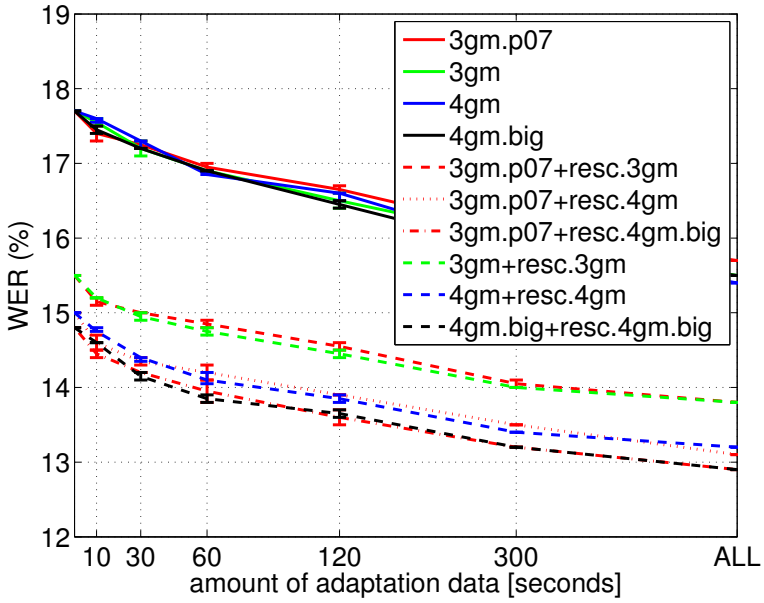
Since our approach relies on the adaptation targets obtained with a first-pass decoding, we investigated the extent to which differentiable pooling methods are sensitive to the quality of the adaptation targets. In this experiment we explored the differences in adaptation hypotheses resulting from different language models, and assumed that the first pass adaptation data was generated by the speaker independent model that will be adapted. The main results are shown for `Diff-Gauss` in Fig. 6.7 (c) where the solid lines show WERs obtained with a pruned 3-gram LM and different types of adaptation targets resulting from re-scoring the adaptation data with stronger LMs. One can see there is not much difference in adaptation accuracies resulting from different speaker independent hypotheses (the absolute difference in WER due to the quality of adaptation targets is about 3% absolute). This trend holds regardless of the amount of data used for adaptation and the overall findings holds also for `Diff-`L_p and `Diff-`L_2 models (results not reported due to space constraints).

6.5.2.3 Summary of results

Results for the proposed techniques are summarised in Tables 6.5, 6.6, and 6.7 for AMI, TED, and SWBD, respectively. The overall observed trends are as follows: (I) speaker independent pooling models return lower WERs than the baseline ANNs: `Diff-Gauss` $<$ `Diff-`$L_2 \leq$ `Diff-`L_p (although the last two seem to be data-dependent); (II) the pooling models (`Diff-Gauss`, `Diff-`L_2 and `Diff-`L_p) are complementary to both fMLLR and `LHUC` adaptation – as expected, the final gain depends on the degree of data mismatch; (III) one can effectively train speaker independent `Diff-`L_2 models and later alter p in a speaker dependent



(a)



(b)

Figure 6.8: WER(%) for different qualities of adaptation targets for the adapted (a) `Diff-`L_p and (b) `Diff-Gauss` acoustic models. Terms 3gm.p07, 3gm, 4gm and 4gm.big correspond to `3gm-312MW.p07`, `3gm-312MW`, `4gm-312MW` and `4gm-751MW`, respectively. Pruning threshold was set to 10^{-7}. Notation 'LM+resc.LM' means the first pass adaptation targets were obtained with 'LM' (possibly re-scored with this LM) and the corresponding adapted model second pass hypotheses were finally re-scored with 'resc.LM'. See text for further description.

Table 6.5: WER(%) on AMI - Individual Headset Microphones and AM trained on FBANK features

Model	`amidev`	`amieval`
ANN	26.8	29.1
+`LHUC`	25.6	27.1
`Diff-Gauss`	26.7	29.0
+ Update μ, β	26.0	27.3
++`LHUC`	25.7	27.0
`Diff-`L_2	26.1	28.5
+ Update p	25.5	26.9
++`LHUC`	25.3	26.7
`Diff-`L_p	25.4	27.6
+ Update p	24.7	25.8
++`LHUC`	24.7	25.6

manner; (IV) the average relative improvement across all tasks with respect to baseline unadapted ANN models were 6.8% for `Diff-Gauss`, 9.1% for `Diff-`L_2 and 10.4% for `Diff-`L_p; and (V) when comparing `LHUC` adapted ANN to `LHUC` adapted differentiable pooling models, the relative reductions in WER for the pooling models were 2%, 3.4% and 4.8% for `Diff-Gauss`, `Diff-`L_2 and `Diff-`L_p, respectively.

6.6 Summary and Discussion

We have proposed the use of differentiable pooling operators with DNN acoustic models to perform unsupervised speaker adaptation. Differentiable pooling operators offer a relatively-low dimensional set of parameters which may be adapted in a speaker-dependent fashion.

We investigated the complementarity of differentiable pooling adaptation with two other approaches – model-based `LHUC` adaptation and feature-space fMLLR adaptation. We have not performed an explicit comparison with an i-vector approach to adaptation. However, some recent papers have compared i-vector adaptation with either `LHUC` and/or fMLLR on similar data which enables us to make indirect comparisons. For example, Samarakoon and Sim [2016] showed that speaker-adaptive training with i-vectors gives a comparable results to test-only

`LHUC` using TED data, and Miao et al. [2015] suggested that `LHUC` is better than a standard use of i-vectors (as in [Saon et al., 2013]) on TED data, with a more sophisticated i-vector post-processing needed to equal `LHUC`. Since the proposed `Diff-`L_p and `Diff-Gauss` techniques resulted in WERs that were at least as good as `LHUC` (and were found to be complementary to fMLLR) we conclude that the proposed pooling-based adaptation techniques are competitive.

We leave as a further work an extension of the proposed techniques to speaker adaptive training (SAT) [Anastasakos et al., 1996, Gales, 2000], for example in a similar spirit as proposed in the context of `SAT-LHUC` in the previous chapter. In addition it would be interesting to investigate the suitability of adapting pooling regions in the framework of sequence discriminative training [Povey, 2003, Kingsbury, 2009, Vesely et al., 2013a]. Our experience of `LHUC` in this framework, presented in previous chapter, together with the observation that the pooling models are not prone to over-fitting in the case of small amounts of adaptation data, suggests that adaptation based on differentiable pooling is a promising technique for sequence trained models.

Table 6.6: Summary WER(%) results on TED test sets from IWSLT12 and IWSLT13 evaluations.

Model	dev2010	tst2010	tst2011	tst2013
Baseline models				
ANN	15.4	15.0	12.1	22.1
+LHUC	14.5	12.8	11.0	19.2
+fMLLR	14.5	12.9	10.9	20.8
++LHUC	14.1	11.8	10.3	18.4
Diff-Gauss models				
Diff-Gauss	15.4	14.6	11.9	21.8
+ Update μ, β	14.5	12.8	11.2	19.5
++LHUC	14.1	12.5	10.8	18.7
+fMLLR	14.6	13.1	10.9	21.1
++ Update μ, β	14.3	12.4	10.7	19.4
+++LHUC	14.1	12.1	10.5	18.9
Diff-L_2 models				
Diff-L_2	15.0	14.6	11.8	21.7
+ Update p	14.1	12.6	11.0	18.5
++LHUC	13.9	12.3	10.8	18.1
Diff-L_p models				
Diff-L_p	14.9	14.5	11.7	21.6
+ Update p	14.2	12.5	10.8	18.4
++LHUC	14.0	12.2	10.6	17.9
+fMLLR	14.0	12.5	10.6	20.3
++ Update p	13.7	11.5	10.0	18.0
+++LHUC	13.4	11.4	9.8	17.6

Table 6.7: Summary WER(%) results on Switchboard `eval2000`

Model	`eval2000`		
	SWB	CHE	TOTAL
Baseline models			
ANN	15.8	28.4	22.1
+`LHUC`	15.4	27.0	21.2
+fMLLR	14.3	26.1	20.3
++`LHUC`	14.2	25.6	19.9
`Diff-Gauss` models			
`Diff-Gauss`	15.1	27.8	21.4
+ Update μ, β	14.8	26.6	20.7
++`LHUC`	14.6	26.2	20.4
+fMLLR	14.4	26.1	20.3
++ Update μ, β	14.3	25.5	19.9
`Diff-`L_2 models			
`Diff-`L_2	14.9	28.0	21.3
+ Update p	14.2	26.0	20.1
++`LHUC`	14.2	25.9	20.1
+fMLLR	13.9	25.5	19.7
++ Update p	13.5	24.9	19.2
`Diff-`L_p models			
`Diff-`L_p	14.8	28.0	21.3
+ Update p	14.2	26.0	20.1
++`LHUC`	14.1	25.9	20.0
+fMLLR	13.7	25.3	19.5
++ Update p	13.5	24.6	19.0

Part III

Low-resource Acoustic Modelling

Chapter 7

Multi-task Acoustic Modelling and Adaptation

This chapter is based on [Swietojanski, Bell, and Renals, 2015] published at ISCA Interspeech. The chapter proposes a multi-task adaptation approach in which mono-phone targets are used to adapt a context-dependent tied-states layer, we also present a number of experiments on low-resource acoustic modelling.

7.1 Introduction

Modelling context-dependent (CD) phones using tied-state clustered trees, initially proposed by Young and Woodland [1994], has been a cornerstone of acoustic modelling for more than two decades, providing a flexible data-driven framework for managing the trade-off between the amount of training material and the final size of the model. Combining this technique within hybrid ANN-HMM framework was one of the major factors in the recent success of ANNs for acoustic modelling.

Despite its widespread and successful use, the optimal clustering for GMM-based systems is often suboptimal for ANNs [Wang and Sim, 2014, Bacchiani and Rybach, 2014]. Under data-constrained conditions some additional initialisation techniques [Dahl et al., 2012, Seide et al., 2011, Zhang and Woodland, 2014b, Swietojanski et al., 2012, Miao and Metze, 2013] need to be applied to fully utilise large CD trees and acoustic adaptation of such models is harder with small amounts of data due to sparsity of adaptation targets. To address some of those issues we propose a structured output layer – an approach that allows the

optimisation and prediction of CD and context-independent (CI) targets jointly, with an explicit dependence of CD targets on CI targets. This makes it possible to use CI predictions at test time as well as learning a more difficult task in combination with an easier one and adapt the CD model with auxiliary monophone targets.

7.2 Structured Output Layer

We build our model based on a multi-task (MT) learning approach of Caruana [1997] and its applications to robust [Parveen and Green, 2003] and cross-lingual [Huang et al., 2013b, Heigold et al., 2013, Ghoshal et al., 2013] acoustic modelling, where the hidden representation is shared and jointly optimised across tasks. In this chapter we are concerned with multi-task training and adaptation within a single language. The choice of an auxiliary task was inspired by the work of Zhang and Woodland [2014b] who found the use of CI targets for layer-wise discriminative pre-training followed by CD fine-tuning leads to models that better generalize, and, due to the low dimensionality of the CI task, are also faster to pre-train. The idea of layer-wise pre-training itself was proposed by Bengio et al. [2007] and was further explored in acoustic modelling for speech recognition by Seide et al. [2011]. However, in [Seide et al., 2011], contrary to [Zhang and Woodland, 2014b], pre-training and fine-tuning relied on the same context-dependent task. More recently Bell and Renals [2015] extended the CI-based initialisation technique to multi-task training where both context-independent and context-dependent targets are jointly trained. All these methods implicitly implement a form of *curriculum learning* [Bengio et al., 2009] where a lower entropy task (with respect to the complexity of classification task or the number of the optimised weights used for intermediate predictions) is employed to iteratively place some relevant prior on the parameters: for example, by forcing the model to predict simpler (but related) concepts first, or using initially fewer parameters which are then expanded as the training progresses.

In this chapter we further extend [Zhang and Woodland, 2014b, Bell and Renals, 2015] by using the CI layer not only at the (pre-)training stage but also to compute CD outputs at run-time and use it for multi-task unsupervised speaker adaptation – the structured output layer (SOL). The SOL estimates the CI outputs m_t as an auxiliary task – (7.1) and (7.2). In the original multitask

formulation, the CD outputs q_t would be estimated independent of the CI outputs at runtime – (7.3) and (7.4) – whereas using the SOL, the CD outputs are given by (7.5) and (7.6). If $\mathbf{a}_m$ represents the CI layer activations, and $\mathbf{a}_s$ and $\mathbf{a}_{sm}$ represent the CD layer activations with and without dependency on the the CI layer, then we have:

$$\mathbf{a}_m = \left(\mathbf{M}^\top \mathbf{x}_t + \mathbf{m}\right) \tag{7.1}$$

$$P(m_t|\bar{\mathbf{O}}_t) = \text{softmax}(\mathbf{a}_m) \tag{7.2}$$

$$\mathbf{a}_s = \left(\mathbf{S}^\top \mathbf{x}_t + \mathbf{b}\right) \tag{7.3}$$

$$P(q_t|\bar{\mathbf{O}}_t) = \text{softmax}(\mathbf{a}_s) \tag{7.4}$$

$$\begin{aligned}\mathbf{a}_{sm} &= \left(\mathbf{S}^\top \mathbf{x}_t + \mathbf{C}^\top \psi\left(\mathbf{a}_m\right) + \mathbf{b}\right) \\ &= \left(\mathbf{a}_s + \mathbf{C}^\top \psi\left(\mathbf{a}_m\right)\right)\end{aligned} \tag{7.5}$$

$$P(q_t|\bar{\mathbf{O}}_t) = \text{softmax}(\mathbf{a}_{sm}), \tag{7.6}$$

where $\mathbf{o}_t$ is the acoustic input. The SOL layer, depicted in Fig 7.1, is then composed of parameters $\boldsymbol{\theta}_{SOL} = \{\mathbf{S}, \mathbf{M}, \mathbf{C}, \mathbf{b}, \mathbf{m}\}$, where $\mathbf{S} \in \mathbb{R}^{X \times S}$ and $\mathbf{b} \in \mathbb{R}^S$ represent hidden to CD weight matrix and bias, respectively. $\mathbf{M} \in \mathbb{R}^{X \times M}$ and $\mathbf{m} \in \mathbb{R}^M$ are for hidden to CI targets while $\mathbf{C} \in \mathbb{R}^{M \times S}$ are the CI to CD connection weights, allowing us to use the easier monophone prediction task when deciding on the (harder) context-dependent tied-state both at training and run-time. ψ is the nonlinearity used for the activations of the CI layer in the SOL. The remaining part of the model follows the usual structure with L hidden layers $\{\mathbf{h}^1, \ldots, \mathbf{h}^L\}$. In the remainder of this chapter, we will be focused mostly on the SOL layer itself, rather than the model as a whole. As such, we introduce an auxiliary variable $\mathbf{x}_t \in \mathbb{R}^X$ which denotes the vector of top hidden layer activations at a time t, or when considering a mini-batch of examples, $\mathbf{x}_t$ becomes $\mathbf{x} \in \mathbb{R}^{X \times B}$, where B is the mini-batch size.

The multi-task optimisation objective may be expressed as a weighted average of the sub-objectives of N separate tasks:

$$\mathcal{F} = \sum_{i=1}^{N} \alpha_i \mathcal{F}_i \tag{7.7}$$

In this case, the global cost has two components; CD ($\mathcal{F}_s = -\sum_t \log P\left(q_t|\bar{\mathbf{O}}_t; \boldsymbol{\theta}_s\right)$) and CI ($\mathcal{F}_m = -\sum_t \log P\left(m_t|\bar{\mathbf{O}}_t; \boldsymbol{\theta}_m\right)$) and both are optimised by gradient descent on the negative log likelihood over T training examples. Note that we obtain

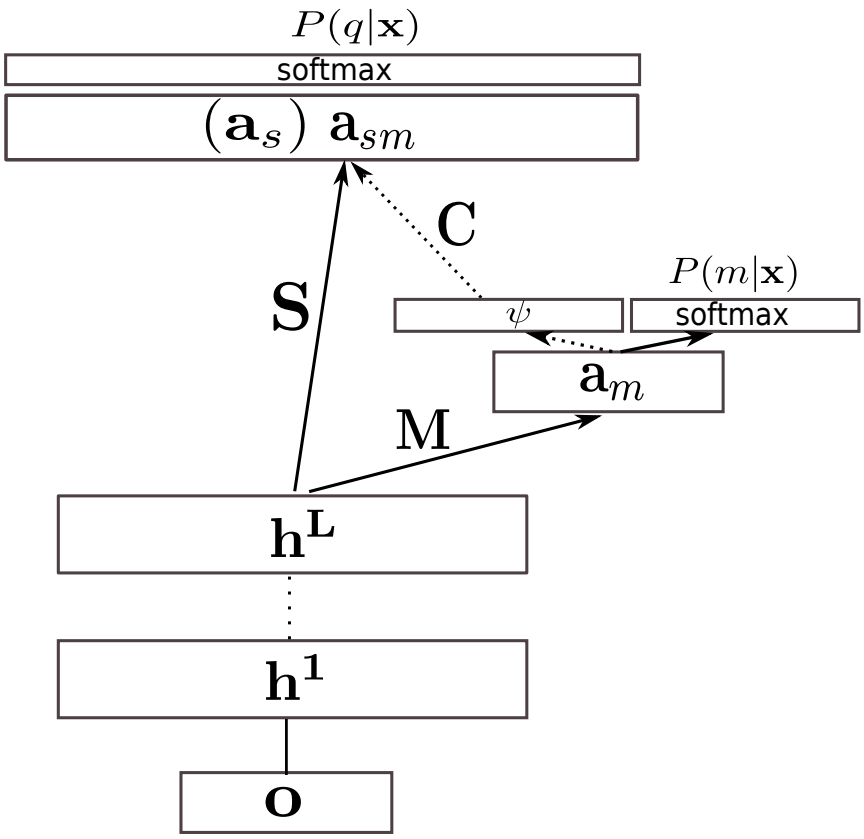


Figure 7.1: The model with the structured output layer (SOL). $P(q|\mathbf{x})$ can be computed with or without a dependendency on the mononphone layer to compute either $\mathbf{a}_{sm}$ (7.5) or $\mathbf{a}_s$ (7.3).

both predictions in parallel in a single forward-pass which is different from Bell and Renals [2015] and from the multi-task framework in general, where the tasks are usually treated as independent and processed sequentially. Effectively, the gradients used to update the parameters are expressed as the weighted average (using hyper-parameter α) of the two tasks with the kth parameter's gradient taking the following form:

$$\frac{\partial \mathcal{F}}{\partial \boldsymbol{\theta}^k} = -\sum_t^T \Big[(1-\alpha) \frac{\partial}{\partial \boldsymbol{\theta}_s^k} \log P\left(q_t | \bar{\mathbf{O}}_t; \boldsymbol{\theta}_s\right) + \alpha \frac{\partial}{\partial \boldsymbol{\theta}_m^k} \log P\left(m_t | \bar{\mathbf{O}}_t; \boldsymbol{\theta}_m\right) \Big] \tag{7.8}$$

Given that $\boldsymbol{\theta}$ includes all the model parameters (including those in the hidden layers), the task-specific parameter subsets are defined as $\boldsymbol{\theta}_s = \boldsymbol{\theta} \setminus \{\mathbf{M}, \mathbf{m}\}$ and $\boldsymbol{\theta}_m = \boldsymbol{\theta} \setminus \{\mathbf{S}, \mathbf{C}, \mathbf{b}\}$ for $\mathcal{F}_s$ and $\mathcal{F}_m$, respectively. In practice, to perform updates, we simply set unrelated gradients (with respect to the given cost) to zero when computing final partial derivatives in (7.8), for example, we set $\partial \mathcal{F}_m / \partial \mathbf{S} = 0$ and

scale the corresponding learning rate for $\partial\mathcal{F}_s/\partial\mathbf{S}$ by $1/(1-\alpha)$. Likewise, for $\mathcal{F}_m$ we set $\partial\mathcal{F}_s/\partial\boldsymbol{\theta}_m = 0$ and scale the CI learning rate by $1/\alpha$.

Depending on our assumptions, the back-propagation of the CD errors may also influence the parameters on the CI path, including the $\mathcal{F}_m$ classification layer. We consider four scenarios:

1. Gradients of $\mathcal{F}_s$ on the "monophone" path are truncated after $\mathbf{C}$ and the back-propagated errors from the cost $\mathcal{F}_s$ to the lower layers are:

$$\frac{\partial\mathcal{F}_s}{\partial\mathbf{x}_t} = \frac{\partial\log P\left(q_t|\bar{\mathbf{O}}_t;\boldsymbol{\theta}_s\right)}{\partial\mathbf{a}_s}\frac{\partial\mathbf{a}_s}{\partial\mathbf{x}_t}$$

2. Gradients flow through $\mathbf{M}$ down to the lower layers, but $\mathbf{M}$ and $\mathbf{m}$ are considered constant relative to $\mathcal{F}_s$, so $\{\partial\mathcal{F}_s/\partial\mathbf{M}, \partial\mathcal{F}_s/\partial\mathbf{m}\} = 0$ and the error signals are:

$$\frac{\partial\mathcal{F}_s}{\partial\mathbf{x}_t} = \frac{\partial\log P\left(q_t|\bar{\mathbf{O}}_t;\boldsymbol{\theta}_s\right)}{\partial\mathbf{a}_s}\frac{\partial\mathbf{a}_s}{\partial\mathbf{x}_t} + \frac{\partial\log P\left(q_t|\bar{\mathbf{O}}_t;\boldsymbol{\theta}_s\right)}{\partial\mathbf{a}_{sm}}\frac{\partial\mathbf{a}_{sm}}{\partial\psi}\frac{\partial\psi}{\partial\mathbf{a}_m}\frac{\partial\mathbf{a}_m}{\partial\mathbf{x}_t} \quad (7.9)$$

3. $\mathcal{F}_S$ influences all dependent parameters, so the back-propagation is as in point 2 above, but partial derivatives $\partial\mathcal{F}_s/\partial\mathbf{M}$ and $\partial\mathcal{F}_s/\partial\mathbf{m}$ are non-zero and used to update $\mathbf{M}$ and $\mathbf{m}$ in eq. (7.8).

4. $\mathbf{C}$ in not learned jointly in MT learning but is added at a post-processing stage and fine-tuned given the predictions for $P\left(q_t|\bar{\mathbf{O}}_t;\boldsymbol{\theta}_s\right)$ and $P\left(m_t|\bar{\mathbf{O}}_t;\boldsymbol{\theta}_m\right)$

The model with a SOL layer exhibits the advantages of classic single-language multi-task approaches [Bell and Renals, 2015, Chen et al., 2014, Seltzer and Droppo, 2013] – its hidden representation is shared across the tasks, so the resulting features are less prone to over-fitting and, as a result, should yield a better generalisation.

Similar approaches to better handle the sparsity of CD states in small amounts of adaptation data with auxiliary targets have been proposed by Price et al. [2014] and Huang et al. [2015a]. Price et al. [2014] proposed to add an additional monophone layer after the one modelling CD states, and then adapted the model with back-propagation and errors computed for the monophone layer. This hierarchical approach is somewhat similar to the concept of adaptation with a linear output [Li and Sim, 2010]. The work of Huang et al. [2015a], which was published independently to ours at the same conference, focuses primarily on multi-task

adaptation. They investigated different types of targets for a secondary adaptation task, including monophones (as in this work) and also clustered the initial CD states to form an auxiliary adaptation characterised by lower sparsity compared to monophones. Moderate gains were obtained with MT adaptation and monophone targets as an auxiliary task.

The other potential advantage comes from a modelling perspective: it is well known that a perceptron (or a logistic classification layer) can only solve linearly separable problems, with Exclusive Or (XOR) being an infamous example of a non-linearly separable problem [Minsky and Papert, 1969]. It is also clear that the transformed acoustic features in the top hidden layer retain highly non-linear characteristics (this can be seen by an error analysis). The well known solution for the "perceptron problem" is an extra intermediate layer connecting the inputs with the outputs [Rumelhart et al., 1986], or in a even simpler scenario, an extra unit describing the relation between the inputs and sending the outcome to the output unit. The latter case is what an auxiliary layer can do in our model, projecting the activations onto CI space, based on which the CD layer can additionally partition the CD space using CI predictions.

The idea of auxiliary targets has been investigated as a "local" coordinate optimisation system [Carreira-Perpiñán and Wang, 2014], where a long chain of back-propagation through many layers is replaced by a shallow sequence of layer-oriented objectives.

7.3 Experiments

We carried our experiments using the TED talks corpus and the Switchboard corpus of conversational telephone speech, described in Sections 4.2 and 4.4, respectively. For Switchboard we exactly follow the recipe described in 4.4 and used in Chapters 5 and 6.

For TED talks we primarily work on 143 hours of training data as used in the previous chapters. For the purpose of this work, we additionally sub-sample random subsets of 10 and 30 hours of training material to simulate low-resource conditions. We performed most experiments on the 30 hours split, reporting results for the most promising configurations on 10 hours and the full 143 hours. For the CI task we use 186 position-dependent phones, and in some control experiments we use 45 monophones. For data constrained scenarios (10 and 30 hours)

Table 7.1: WER(%) results on `tst2010` set. Models trained on 30 hour data-split with MT interpolation hyper-parameter $\alpha = 0.3$.

	Model	`tst2010`	+`4gm-312MW`
S1	ANN (1k hidden units)	23.1 ± 0.1	19.7
S2	`SOL-ANN` const.$\partial\mathcal{F}_s/\partial\{\mathbf{M},\mathbf{m}\}$	22.8	19.5
S3	`SOL-ANN` $\partial\mathcal{F}_s/\partial\boldsymbol{\theta}_m$	21.9 ± 0.2	18.7
S4	`SOL-ANN` + PI Monophones	22.7	19.4
S5	`SOL-ANN` + Retrained CD	22.5	19.2
S6	ANN (2k hidden units)	22.6	19.2
S7	`SOL-ANN` (2k hidden units) $\partial\mathcal{F}_s/\partial\boldsymbol{\theta}_m$	21.6	18.5

our models have 6 hidden layers with 1,000 units each. We additionally perform low-rank factorisation of the output CD layer by inserting a linear-bottleneck [Sainath et al., 2013a, Grezl et al., 2007], i.e. our layer becomes $\mathbf{S} = \mathbf{S}_{in} \times \mathbf{S}_{out}$, where $\mathbf{S}_{in} \in \mathbb{R}^{X \times L}$ and $\mathbf{S}_{out} \in \mathbb{R}^{L \times S}$ with L=256.

7.3.1 Structured output layer

In this section we look at different training scenarios for ANN with structure output layer, termed `SOL-ANN`, comparing with a baseline ANN model, with 1,000 hidden units and the low-rank factorisation of the CD output layer. The baseline results are given in row (S1) of Table 7.1.

We explored the training scenarios outlined in Section 7.2. We found that both truncation of $\mathbf{C}$ (scenario 1) and optimising $\mathbf{C}$ as a post-processing step (scenario 4) resulted in very high frame error rates in comparison with the baseline. Row (S2) gives word error rates (WERs) for the case where the CD cost is not used to update $\mathbf{M}$ and $\mathbf{m}$; row (S3) shows the opposite scenario indicating that updating the CI-dependent parameters using the $\mathcal{F}_s$ cost yields the lowest WER, 21.9%, a 6% relative improvement over the baseline (p_v<0.001). Row (S4) is a model trained on 45 position-independent phones (compared to the other models utilising 186 position-dependent phones). Model (S5) is built from the hidden representation of (S3) with a new CD regression layer showing that both the SOL layer and multitask training are important.[1] Finally, rows (S6) and (S7)

[1]We did some sanity checks, and the corresponding models (S1) and (S3) with retrained top layers converged to their base model accuracies when all layers were jointly optimised.

Table 7.2: WER(%) on `tst2010` set for different $\mathbf{M}$ to $\mathbf{C}$ ψ activations. The base model is `SOL-ANN` $\partial\mathcal{F}_s/\partial\boldsymbol{\theta}_m$ and MT interpolation hyper-parameter $\alpha = 0.3$

ψ-activation				
`linear`	`softmax`	`sigmoid`	`relu`	`tanh`
21.9	22.2	22.5	22.4	23.1

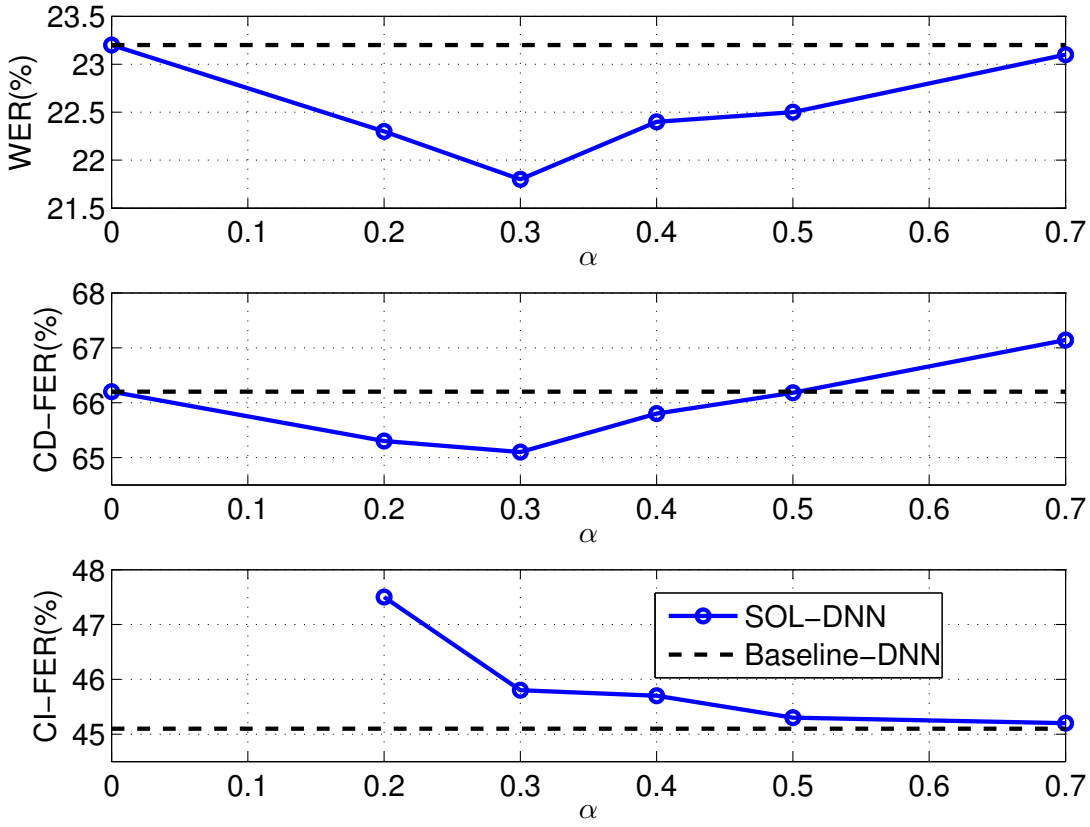


Figure 7.2: WER and FER as a function of MT interpolation hyper-parameter α.

present WERs for larger models showing a gain of over 1% absolute (or 0.7% for re-scored lattices, p_v<0.001) gain for the `SOL-ANN` structure.

Table 7.2 presents WERs for different activation functions (ψ) connecting $\mathbf{M}$ and $\mathbf{C}$ using model (S3) from Table 7.1. The linear connection was found to work best, and in the following part of the chapter we follow the structure and optimisation procedure used to train model (S3).

Figure 7.2 shows WER (on `tst2010`), as well as corresponding CD and CI frame error rates (FER) for different weighting constants α, the best WER results (and also FER for CD task) were obtained with α=0.3.

Finally, Fig 7.3 shows convergence plots of baseline and `SOL-ANN` models (no

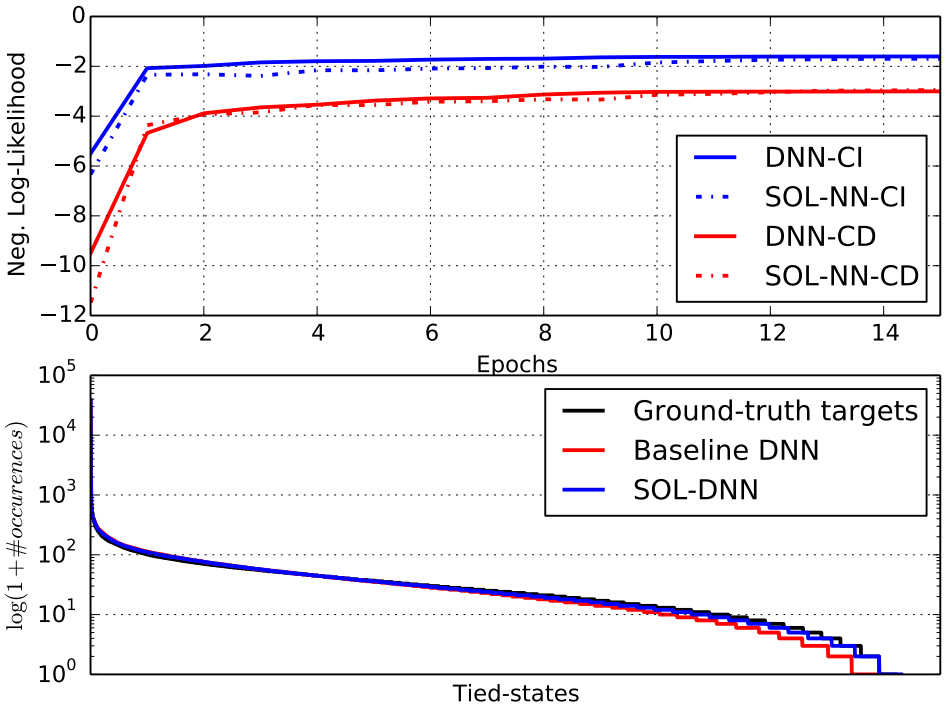


Figure 7.3: Top) Convergence plots for SOL-ANN and baseline CD/CI models models on dev2010. Bottom) Distribution of CD states obtained from ground-truth labels and the estimates of ANN and SOL-ANN models.

significant differences) as well as the predicted distributions of CD states under both models compared with the expected one obtained using ground-truth alignments of dev2010 (all sorted by occurrence frequencies). The SOL-ANN better deals with modelling a tail of a distribution, which could explain why there are only small differences in the log likelihoods but significant reductions in word error rates.

7.3.2 Multi-task adaptation

In this section we investigate the feasibility of using the CI targets to aid unsupervised two-pass adaptation. Our motivation is that CI modelling is usually characterised by a lower frame error rate, and at the same time there is less sparsity in the distribution of CI targets, given the same amount of adaptation data, hence potentially better suited for adaptation compared to a CD-only objective.

We adapt the speaker-independent models with learning hidden unit con-

Table 7.3: Detailed results on tst2010 and unsupervised adaptation with multi-task LHUC using auxiliary targets on 30 hours models (4gm-312MW LM). The scenario when $\alpha = 0$ corresponds to pure CD adaptation. Likewise $\alpha = 1$ is pure CI adaptation.

	MT hyper-parameter α				
System	0	0.3	0.5	0.7	1
Baseline	18.6				
Adaptation with 10 seconds per speaker					
+LHUC	18.0	18.0	17.9	18.2	18.45
Adaptation with all speaker's data					
+LHUC	16.0	15.7	15.8	15.7	16.1

tributions (LHUC) described in Chatper 5 with unsupervised adaptation data. We report our control adaptation results for two scenarios, using both a limited amount of 10 seconds of speech per speaker as well as full two pass adaptation. For the 10s scenario we repeated the experiments 5 times, for randomly selected utterances, and report the average WERs.

The results on tst2010 are reported in Table 7.3 showing that around 0.1-0.3% absolute gain was obtained on top of CD-only adaptation for both scenarios. We observe similar gains when adapting models on other test sets. With $\alpha = 0.5$, WER on dev2010 is 0.3% abs. lower for the 10s adaptation scenario. Similarly to tst2010, interpolated ($\alpha = 0.5$) adaptation on dev2010 with the whole speaker's data reduced the WER by 0.2% abs. when compared to CD-only adaptation. On tst2011 adapting with 10s gave smaller reductions for both methods (0.2% abs.) regardless of α; this could be due to the fact tst2011 is better matched to training conditions and benefits less from adaptation. Figure 7.4 shows CD frame error rates for different settings of CD/CI interpolation coefficient α.

7.3.3 Full TED and Switchboard

We also report summary results on the three scenarios for TED (10, 30 and 143 hours of training data) in Table 7.4 using 4gm-751MW LM which is the same with the one used in Chapters 5 and 6. In particular, we compare multi-task adaptation of SOL-ANN models using LHUC (denoted as LHUC (MT)) with the scenario where LHUC parameters are estimated using only a single CD task. The results are reported in Table 7.4 and show that adapting with LHUC (MT), on

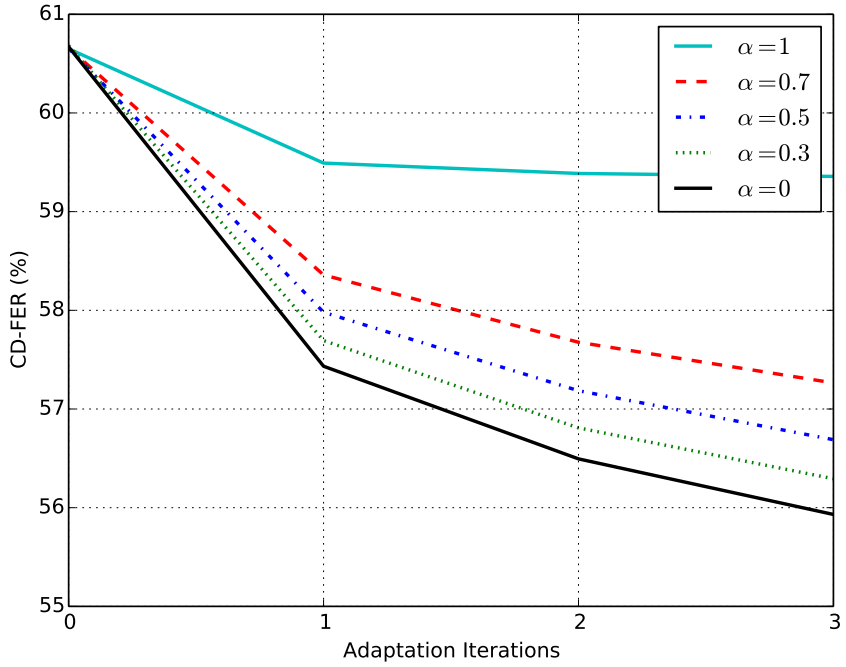


Figure 7.4: Context-dependent frame error rates as a function of adapting iterations for different values of MT hyper-parameter α on `tst2010`.

average across all 4 datasets, bring small but consistent gains (p_v<0.01). This is in line with Huang et al. [2015a] where also only moderate gains from MT adaptation were obtained. Multi-task `SOL-ANN` models are also much better in speaker-independent under-resourced acoustic modelling offering consistent accuracy improvements over the baseline ANN models (p_v<0.001). For larger `SOL-ANN` models and more training data the gains from MT training diminish in both speaker-independent and speaker-adaptive settings, similar conclusion for MT training was also found in [Siohan, 2016]. Some additional insights are given in [Bell et al., 2016], which suggest the importance of sequential, rather than parallel, presentation of the auxiliary tasks.

On Switchboard (Table 7.5) the `SOL-ANN` model reduced the WER on the Switchboard (SWB) part while at the same time increasing the WER on the CallHome (CHE) test data, an increase of 0.2% WER on average compared with ANN-HMM baseline. This indicates the SOL layer might overfit conditions of training data, hence improvement in WERs on matched conditions of SWBD part of test set, but deterioration in WER on CHE part.

Table 7.4: Summary results on the remaining TED test-sets and different amounts of training material and LHUC adaptation and 4gm-751MW language model. LHUC (MT) stands for multi-task adaptation with $\alpha = 0.5$.

	WER (%)			
System	dev2010	tst2010	tst2011	tst2013
10 hour				
ANN	23.1	24.0	19.4	34.7
SOL-ANN	21.6	22.5	18.0	32.8
+LHUC	19.7	19.4	16.6	28.7
+LHUC (MT)	19.5	19.0	16.4	28.6
30 hour				
ANN	19.8	19.7	15.8	28.9
SOL-ANN	19.0	18.4	15.4	27.6
+LHUC	17.3	15.8	13.3	23.8
+LHUC (MT)	17.0	15.7	13.0	23.8
143 hour				
ANN	15.4	15.0	12.1	22.1
SOL-ANN	15.5	15.1	12.0	22.1
+LHUC	14.5	12.9	10.9	19.1
+LHUC (MT)	14.5	12.8	10.9	19.2

Table 7.5: WER(%) on Switchboard Hub00

	Hub5'00		
Model	SWB	CHE	TOTAL
ANN	15.8	28.4	22.1
SOL-ANN	15.6	28.9	22.3

7.4 Summary

In this chapter we have proposed a structured output layer, an approach in which an auxiliary (context-independent) task is used as a regulariser during training but also as an auxiliary predictor in deriving context-dependent tied states for decoding. We have investigated various training strategies for this technique, and have shown that SOL-ANN approach is an effective way of addressing an issue of unsupervised adaptation with sparse data.

Chapter 8

Unsupervised multi-lingual knowledge transfer

This chapter is based on [Swietojanski, Ghoshal, and Renals, 2012] published at IEEE Spoken Language Technology Workshop (SLT) and concerns multi-lingual knowledge transfer using restricted Boltzmann machines.

8.1 Introduction

In this chapter we are concerned with building acoustic models with limited amounts of transcribed audio. We assume that we have untranscribed audio in the chosen language, as well as in other languages. The key question that we address is how to usefully employ this untranscribed acoustic data for speech recognition of the target language. We consider this in the context of ANN acoustic models for the target language, which can take advantage of the untranscribed audio using unsupervised pretraining techniques. We use layer-wise restricted Boltzmann machine (RBM) initialisation of an ANN [Hinton et al., 2006], an unsupervised procedure, in which a deep generative model of the acoustic data is estimated and used to initialise the weights of the ANN, which are then refined using supervised training on transcribed acoustic data in the target language. The generative model may be of acoustics in the same language (in-domain) or a different language (out-of-domain). Through these experiments we aim to develop a better understanding of cross-lingual knowledge transfer, as well as unsupervised pretraining. In this chapter we use the ANNs in both tandem and hybrid configurations.

8.2 Review of low-resource acoustic modelling

In cross-lingual speech recognition, knowledge from one or more languages is used to improve speech recognition for a target language that is typically low-resourced. A number of techniques for cross-lingual acoustic modelling have been published including the use of global phone sets [Schultz and Waibel, 2001a,b] and multilingual ANN posterior features for tandem GMM-HMM systems where the posterior is estimated for either context-independent phones [Grézl et al., 2011, Thomas et al., 2012b,a, Lal and King, 2013], context-dependent tied states using an ANN with a bottleneck layer [Knill et al., 2013] or articulatory features [Lal and King, 2013]. In the spirit of the hybrid approach, one can use multi-lingual ANN posteriors to model HMM state distributions using the Kullback-Liebler hidden Markov model (KL-HMM) approach [Imseng et al., 2012]. GMM-HMM systems may be improved by subspace Gaussian mixture models (SGMMs) with a shared multilingual phonetic subspace [Burget et al., 2010, Lu et al., 2014]. There also exist zero-resource approaches relying on cross-lingual bootstrapping with unsupervised training of the target language [Vu et al., 2011]. These approaches rely on transcribed audio data for building automatic speech recognition (ASR) systems in some source languages that may or may not be linguistically related to the target language. These approaches assume that only a small volume of transcribed target language audio is available, and in some cases the target language audio is assumed to be entirely untranscribed [Schultz and Waibel, 2001b, Vu et al., 2011, Saiko et al., 2014].

Prior to 2012, when the work in this chapter was performed, cross-lingual acoustic modelling had mainly focussed on tandem approaches that require transcribed data in the source language(s). A common approach relied on a direct use of posterior features obtained from a source language ANN [Stolcke et al., 2006], the use of cross-lingual bottleneck features [Grézl et al., 2011], training/initialising a neural network using transcribed source language acoustics, then retraining the network with transcribed target language acoustics, using a phoneset mapping where necessary [Thomas et al., 2010, 2012a], and posterior features derived from networks trained to estimate articulatory features [Çetin et al., 2007, Lal, 2011]. To the best of our knowledge, [Swietojanski et al., 2012] was the first work where unlabelled acoustic data from a different language is successfully used to improve speech recognition accuracy. A similar findings to ours, but based on initialisation

using stacked auto-encoders were later also reported by Gehring et al. [2013].

There has been work investigating the applicability of supervised cross-lingual knowledge transfer for hybrid systems. The most successful approach seems to be multi-task joint training of hybrid acoustic models using language-dependent output layers with a shared hidden representation [Heigold et al., 2013, Huang et al., 2013b]. Those approaches are conceptually very similar to the work of Thomas et al. [2012a]. Ghoshal et al. [2013] proposed a *hat-swapping* approach in which hidden representation is shared and sequentially optimised using multiple languages - this technique may be considered as a cross-lingual initialisation, similar to the work presented in this chapter, but in a supervised manner.

Low-resource acoustic modelling may be also considered as an adaptation problem, i.e. given a reliably trained ANN how to adapt it to a target language. There have been a number of techniques investigating this front which overlap with the solutions proposed for acoustic adaptation, for example, Motlicek et al. [2015] investigated the use of language-dependent layers, and [Mohan and Rose, 2015] additionally factorised the layer of low-resource target language to further limit the number of parameters. Likewise MLLR, CMLLR and MAP techniques, classically developed for adaptation in GMM-HMM based systems, have been applied to cross– and multi–lingual acoustic modelling [Bub et al., 1997, Nieuwoudt and Botha, 2002, Zgank et al., 2003].

Perhaps the most obvious way to deal with the low-resource scenario is by transforming it to a rich-resource one. If there are some initial resources to build a seed system one can obtain more supervision material using this system to transcribe more data and hence build a better system. Those techniques are often refereed to as semi-supervised training and are applicable to both mono— and multi–lingual settings [Vesely et al., 2013b, Motlicek et al., 2015]. In fact this is yet another link between adaptation and tackling under-resource scenarios, as many adaptation techniques (including the ones proposed in Chapters 5 and 6) rely on two-pass systems to obtain adaptation targets. The semi-supervised training approaches scale this approach up to full system re-training. Also, industrial systems mostly rely on semi-supervised training, often using some confidence measures to select sufficiently reliable training segments.

8.3 ANNs and restricted Boltzmann machines

Training deep networks directly results in a difficult optimization problem when small amounts of supervised data are available and an unsupervised pretraining phase using greedy layer-wise training of RBMs [Hinton et al., 2006] or stacked autoencoders [Bengio et al., 2007] has been shown to give good results. More recently, supervised layer-wise training with early stopping was shown to achieve comparable or better results than unsupervised pretraining on a relatively large speech recognition task [Seide et al., 2011]. For our investigation of unsupervised cross-lingual pretraining, RBMs were a natural first choice due to their previous successful application in speech recognition [Mohamed et al., 2012, Dahl et al., 2012].

RBMs are bipartite undirected graphical models, with a set of nodes corresponding to observed random variables (also called visible units) and a set of nodes corresponding to latent random variables (or hidden units), that only allow interactions between the two sets of variables (that is, between the visible and hidden units) but not within either set of nodes. The joint probability of the visible units $\mathbf{v}$ and hidden units $\mathbf{h}$ is defined as:

$$P(\mathbf{v}, \mathbf{h}) = \frac{1}{Z_{h,v}} e^{-E(\mathbf{v},\mathbf{h})}, \tag{8.1}$$

where $Z_{h,v} = \sum_{\mathbf{h}} \sum_{\mathbf{v}} e^{-E(\mathbf{v},\mathbf{h})}$ is the normalising partition function. Visible units are real-valued for speech observations and binary-valued otherwise; hidden units are always binary-valued.

In the case of binary visible units, we have a Bernoulli-Bernoulli RBM whose energy function $E(\mathbf{v}, \mathbf{h})$ is defined as:

$$E_{\text{B-B}}(\mathbf{v}, \mathbf{h}) = -\mathbf{v}^\top \mathbf{W}\mathbf{h} - \mathbf{b}_v^\top \mathbf{v} - \mathbf{b}_h^\top \mathbf{h}, \tag{8.2}$$

and for real-valued visible units we use a diagonal covariance Gaussian-Bernoulli RBM whose energy function is given by:

$$E_{\text{G-B}}(\mathbf{v}, \mathbf{h}) = -\mathbf{v}^T \mathbf{W}\mathbf{h} - \frac{1}{2}(\mathbf{v} - \mathbf{b}_v)^T(\mathbf{v} - \mathbf{b}_v) - \mathbf{b}_h^T \mathbf{h}. \tag{8.3}$$

$\mathbf{W}$ is a symmetric weight matrix defining interactions between vectors $\mathbf{v}$ and $\mathbf{h}$ while $\mathbf{b}_v$ and $\mathbf{b}_h$ are additive bias terms for visible and hidden units, respectively. RBM pretraining maximises the likelihood of the training samples using the contrastive divergence algorithm [Hinton et al., 2006]. When multiple layers have to

be initialised the parameters of the given layer are frozen and its output is used as the input to the higher layer which is optimised as a new RBM. This procedure is repeated until the desired number of layers is reached.

When used in tandem configuration [Hermansky et al., 2000], the ANN outputs correspond to posterior probabilities of the context-independent phones in the language (in our case, 44 for German). The outputs are Gaussianized by taking logarithms, decorrelated using principal components analysis (PCA), and concatenated with MFCCs. The PCA step also reduces the dimensionality from 44 to 25 (this guaranteed keeping at least 95% of variance—on average it was 98%), producing a combined 64-dimensional feature for the GMM-HMM acoustic model. In the hybrid setup, the outputs correspond to tied triphone states. Depending on the amount of training data used, the number of tied triphone states may vary from a few hundred to a few thousand (roughly 550 to 2500 in our case). To obtain scaled likelihoods, the posterior probability estimates produced by the network were divided by the prior probabilities [Bourlard and Morgan, 1994].

8.4 Experiments

For testing cross-lingual knowledge transfer in ANNs with RBMs we use the GlobalPhone corpus [Schultz, 2002] described in Section 4.6. Our setup is similar to that reported in [Lu et al., 2014]. We use German as our in-domain language and we simulate different degrees of available resources by selecting random 1 and 5 hour subsets of the total 15 hours of labeled training speech data. When using the 1 and 5 hour subsets, the entire 15 hours of audio from the training set were used for the RBM-based unsupervised pretraining. We contrast this with RBM pretraining using unlabeled acoustic data from three other languages: Portuguese (26 hours), Spanish (22 hours) and Swedish (22 hours), as well as with pretraining using all the languages (85 hours).

8.4.1 Baseline results

Before discussing the results on GlobalPhone, it is important to note that the results reported in various sources (for example, [Schultz and Waibel, 2001a, Grézl et al., 2011, Lu et al., 2011, Lal, 2011]) are not directly comparable. This is pri-

Table 8.1: WER(%) for GMM-HMM systems on GlobalPhone German development set.

Training	Features	Amount of training data		
		15hr	5hr	1hr
ML	MFCC	16.17	18.40	23.11
ML	LDA/MLLT	15.53	18.41	22.31
fBMMI+BMMI	LDA/MLLT	15.19	18.19	21.53

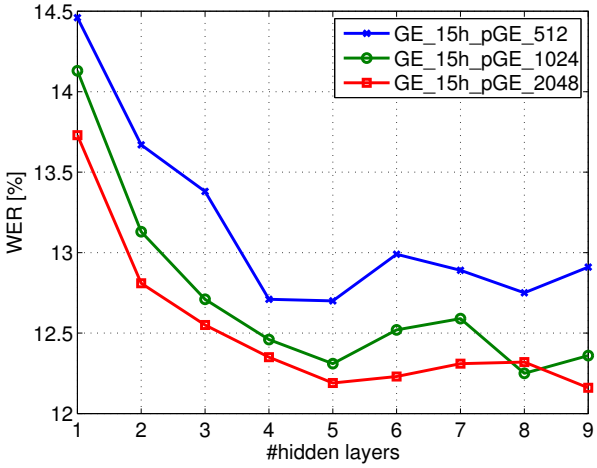
marily because of the differences between LMs, which are much more significant than other differences, such as the use of MFCC vs PLP features. Following previous work [Lu et al., 2011], we use LMs that were included in an earlier release of the corpus, but are not available in later releases. The differences between the results reported here and those in [Lu et al., 2011] are due to the fact that we found it beneficial to interpolate the provided LM with one trained on the training transcripts.

Limited training subsets GE_5h and GE_1h were randomly selected from the complete GE training set (GE_15h) keeping approx. 8 minutes of recorded speech for each of 8 (1h) or 40 (5h) speakers. The number of tied states/GMM components for each of training variants was set to 2564/16, 1322/8 and 551/4. All the reported results were obtained using GlobalPhone GE development or evaluation sets. Ground truth lables used for ANN finetuning were generated separately for each of mentioned datasets based on the corresponding ML MFCC baseline which results on development set are reported in Table 8.1).

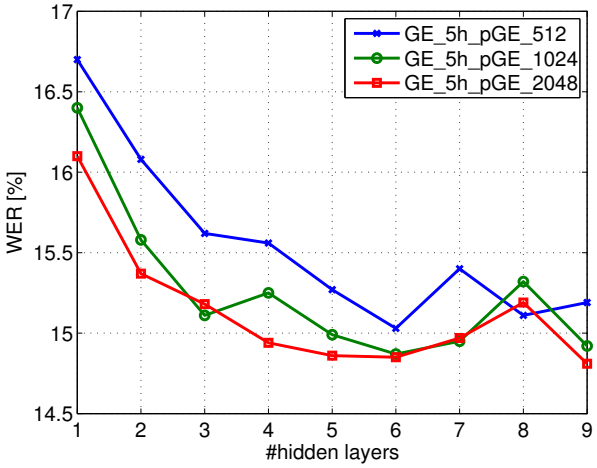
8.4.2 ANN configuration and results

We use 12 PLP coefficients and the energy term appended with the delta and acceleration coefficients for a 39-dimensional acoustic feature vector. The features are globally normalised to zero mean and unit variance, and 9 frames (4 on each side of the current frame) are used as the input to the networks. The choice of PLP features was initially motivated by the desire to have information that is complementary to MFCCs for the tandem configuration.

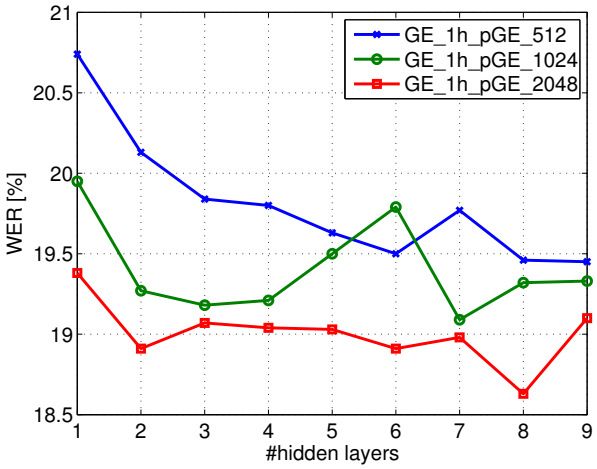
The initial network weights (both for RBM pretraining, and when no pretraining was done) were chosen uniformly at random: $w \sim U[-r, r]$, where $r = 4\sqrt{6/(n_j + n_{j+1})}$ and n_j is the number of units in layer j. We choose the pre-



(a) 15h hybrid system



(b) 5h hybrid system



(c) 1h hybrid system

Figure 8.1: German development set WERs for hybrid systems with different sizes of hidden layers (512, 1024 and 2048 hidden units) for the three training sets.

training hyper-parameters as follows: learning rate for Bernoulli-Bernoulli RBM is 0.08, and for Gaussian-Bernoulli RBM in the input layer it is 0.005. Mini-batch size is 100. Fine-tuning is done using stochastic gradient descent on 256-frame mini-batches and an exponentially decaying schedule, learning as described for TED data in Section 4.2.

In the tandem setup, the networks are up to five layers deep since the tandem systems were not found to improve in terms of WER with deeper networks (Fig. 8.2). The networks have 1024 hidden units per layer, which was found to outperform 512 hidden units and to have similar WER to 2048 hidden units. In contrast, the hybrid system benefits from deeper architectures (Fig 8.3), as well as wider hidden layers with 2048 units, even when fine-tuning using just 1 hour of transcribed speech (Fig. 8.1). This shows the RBM is a strong regulariser and suggest, even for low-resource conditions, it is not of the first importance to appropriately select the total number of ANN parameters. In fact, for 1 hour of data (360 000 training data-points) the best hybrid ANN model tested on unseen conditions had $31.2 \cdot 10^6$ parameters.

We find that the hybrid systems provide lower WER than the corresponding tandem systems. Additionally, and perhaps most importantly, unsupervised RBM pretraining is found to be language-independent. Pretraining is found to be more effective for hybrid systems than for tandem systems, and the effect is most pronounced when the hybrid systems are fine-tuned using limited amounts of transcribed data. In fact, with 1 hour of transcribed speech the hybrid system only outperformed the baseline GMM-HMM system when pretraining was done. However, for both the tandem and hybrid configurations, we see no correlation between the amount of data used for pretraining (which varied between 15 and 85 hours) and the WER obtained by the fine-tuned system.

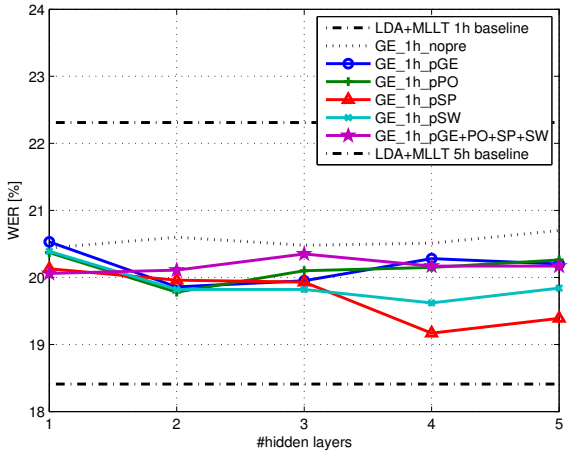
For the different ANN configurations shown in Figures 8.2 and 8.3, we pick the ones with the lowest WER on the development set and use them to decode the evaluation set. The results are shown in Tables 8.2 and 8.3. For the different amounts of training data, the best HMM-GMM, tandem, and hybrid results are summarised in Figure 8.4.

Table 8.2: Tandem system WER(%) results on German eval set

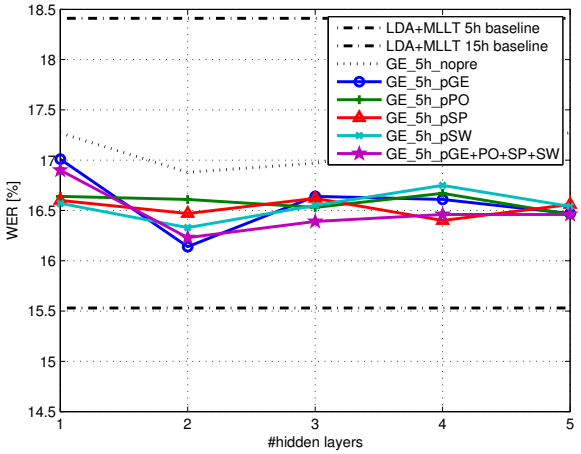
System description	Amount of training data		
	15hr	5hr	1hr
ML using LDA/MLLT	24.53	27.56	34.08
ANN random initialised	22.05	25.10	31.84
ANN pretrained on GE	21.39	24.60	30.91
ANN pretrained on PO	21.21	24.43	31.29
ANN pretrained on SP	21.48	24.23	30.74
ANN pretrained on SW	21.62	24.44	30.52
ANN pretrained on All	21.48	24.49	30.98

Table 8.3: Hybrid system WER(%) results on German eval set

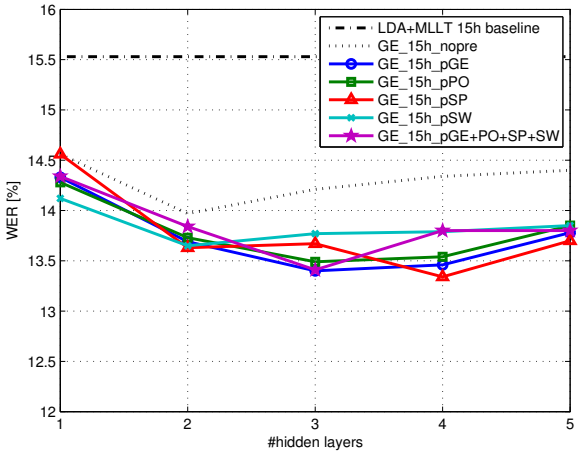
System description	Amount of training data		
	15hr	5hr	1hr
fBMMI+BMMI using LDA/MLLT	24.13	27.08	33.11
ANN random initialised	21.52	25.03	33.54
ANN pretrained on GE	20.09	22.78	28.70
ANN pretrained on PO	20.00	22.44	28.79
ANN pretrained on SP	20.03	22.64	28.40
ANN pretrained on SW	20.20	22.89	28.92
ANN pretrained on All	20.14	22.70	28.72



(a) Tandem German 1h labeled data

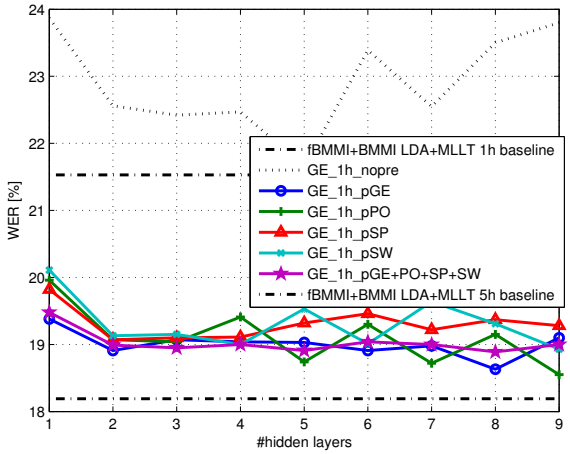


(b) Tandem German 5h labeled data

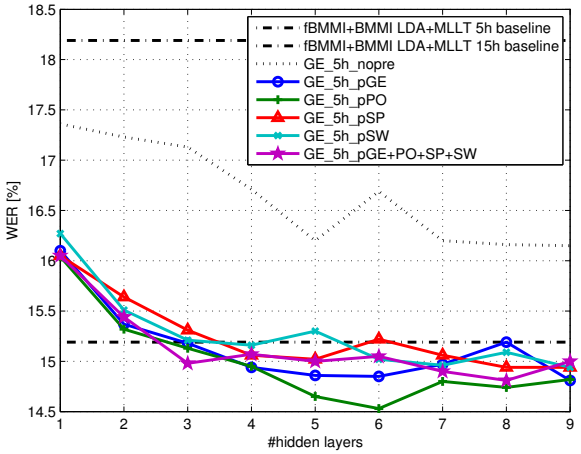


(c) Tandem German 15h labeled data

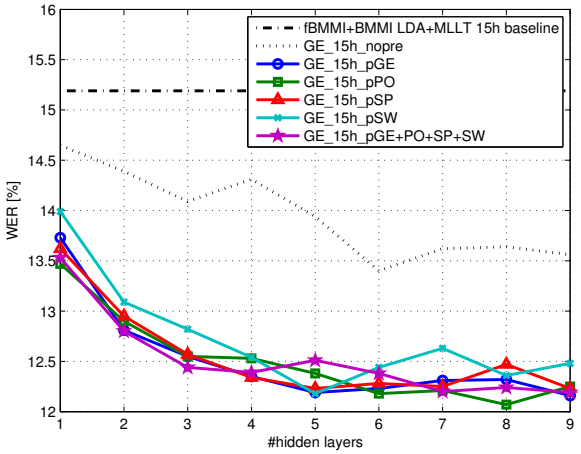
Figure 8.2: Tandem HMM-GMM setup. Results on devset.



(a) Hybrid German 1h labeled data



(b) Hybrid German 5h labeled data



(c) Hybrid German 15h labeled data

Figure 8.3: Hybrid setup. Results on devset.

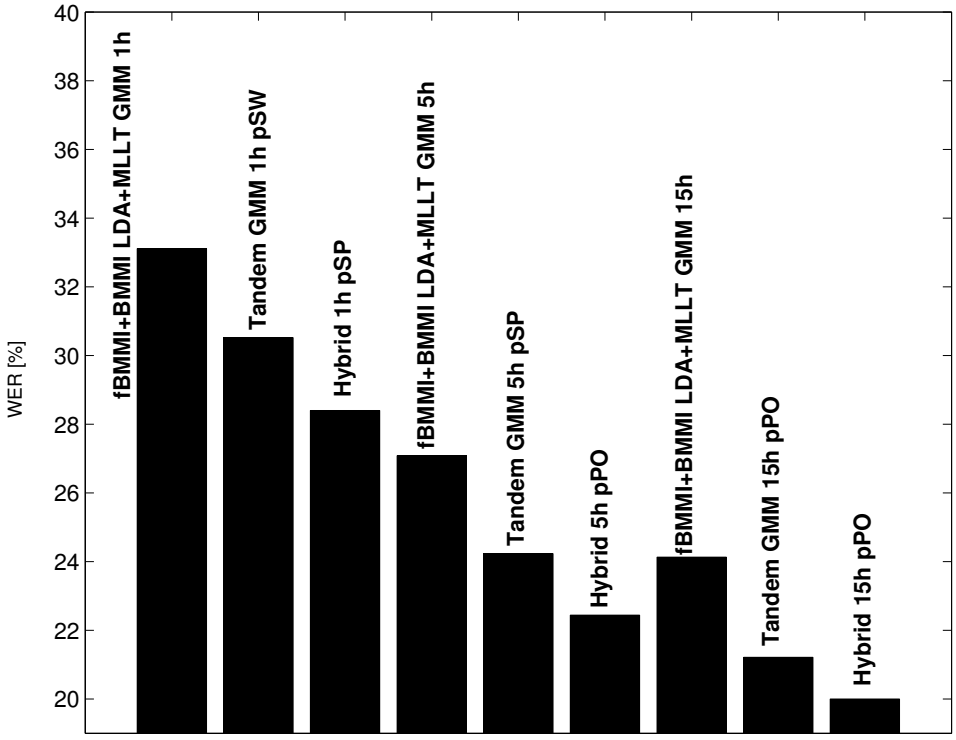


Figure 8.4: A summary of results on the evaluation set. Notation pLANG denotes the model was pre-trained using LANG acoustics in unsupervised manner and then refined with supervision on 1, 5 or 15 hours of transcribed target language speech.

8.5 Summary and Discussion

In this chapter we examined the usability of unlabelled data from one or more languages to improve recognition accuracy of a different, possibly low-resourced, language in a fully unsupervised fashion. The experiments we performed suggest that unsupervised RBM-based initialisation of ANNs is language-independent, allowing hybrid setups to be built from as little as 1 hour of labelled fine-tuning data (there are no statistically significant differences between target language WERs obtained with pretraining on one or many source languages, and all such pretrained acoustic models are significantly better than the randomly initialised models). This simple approach reduces the cost of building an ASR system in a new language by not only requiring less transcribed data, but less amount of data to be collected in the first place.

One may think of cross-lingual speech recognition as an exercise in judicious application of prior knowledge, whether in the linguistic sense of mapping between phonesets, or in the statistical sense of sharing model parameters between languages. Unsupervised pretraining of ANNs fits in this framework. In fact, Erhan et al. [2010] explain unsupervised pretraining as "an unusual form of regularization" that restricts the subsequent supervised (and discriminative) learning to points of the parameter space corresponding to a better generative model of the data. Our results strongly suggest that RBM-based unsupervised pretraining is able to learn characteristics of human speech that are largely language-independent. It is possible, even likely, that this characteristic will be demonstrated by other unsupervised pretraining techniques as well, for example, pretraining using stacked autoencoders [Bengio et al., 2007]. In fact, work of Li et al. [2014b] confirms the language-independence claim in a supervised *teacher-student* setting where assuming well trained *teacher* model, speech data that is later passed through the teacher to aid student training is to a large extent irrelevant (in the paper German data was used to train English student model, with a rather minor drop in the final accuracy).

While pretraining is seen to be language-independent, no clear pattern emerges when going from 15 to 85 hours of data for pretraining. This raises two questions that have not been sufficiently addressed in literature: what makes some data suitable for unsupervised pretraining, and what are sufficient amounts of suitable pretraining data. It is possible that cross-corpus variability offset gains from pre-

training on a mixture of languages; it is also possible that more data is simply not necessary. Better embeddings of the data may be obtained by imparting domain knowledge: for example, pretraining and fine-tuning in a speaker-adaptive fashion may be helpful in a cross-lingual setting. Finally, our approach is complimentary to other cross-lingual ASR approaches, and it is easy to imagine combining cross-lingual ANNs and SGMMs using the tandem approach.

Part IV

Distant Speech Recognition

Chapter 9

Learning Representations from Multiple Acoustic Channels

The content of this chapter is based on [Swietojanski, Ghoshal, and Renals, 2014a] published in IEEE Signal Processing Letters and [Renals and Swietojanski, 2014] published in Workshop on Hands-free Speech Communication and Microphone Arrays (HSCMA). Those papers include some material and ideas from [Swietojanski, Ghoshal, and Renals, 2013b] published in IEEE Automatic Speech Recognition and Understanding Workshop (ASRU).

9.1 Introduction

Distant Speech Recognition (DSR) [Wölfel and McDonough, 2009] remains a significant open challenge. Recognition of speech captured using multiple distant microphones, typically configured in a calibrated array, is a difficult task since the speech signals to be recognised are degraded by the the effects of interference with other acoustic sources and reverberation. A typical approach for DSR as used in NIST RT evaluation campaigns involved two primary components, the use of microphone arrays and multi-stage cross-system adaptation. An excellent example of a tandem GMM-HMM system making use of those components in the meeting transcription task is [Hain et al., 2012].

In this chapter we are primarily concerned with ways of incorporating multiple acoustic channels directly into the ANN model. In particular, we investigate two approaches based on channel concatenation and convolutional neural networks (CNN).

9.2 Review of DSR approaches

Distant conversational speech recognition is highly challenging for several reasons. A typical recording may include multiple overlapping talkers, as well as additional non-speech acoustic sources, and the recording environment may have significant reverberation. During the 1990s a number of pioneering studies investigated the development of DSR systems based on a microphone array (e.g. [Van Compernolle et al., 1990, Adcock et al., 1996, Omologo et al., 1997]), and an evaluation framework for speech recognition based on multichannel recordings of Wall Street Journal sentences [Lincoln et al., 2005] enabled some comparability in this area. In practice, the effect of speaker and channel adaptation has been found to have a much greater effect on speech recognition word error rates, compared with changes to the beamforming algorithm and post-filtering [Zwyssig et al., 2013]. On the other hand, a number of techniques have been developed to address specific challenges such as reverberation and overlapping talkers [Yoshioka et al., 2012, Kumatani et al., 2012].

Most distant speech recognition systems have adopted a two-part architecture in which a microphone array beamforming algorithm is applied to the recorded multichannel speech, followed by conventional acoustic modelling approaches. Good examples of such systems include the AMIDA [Hain et al., 2012] and ICSI/SRI [Stolcke et al., 2008] systems for meeting transcription. Both of these systems process the microphone array signals using a noise-reducing Wiener filter on each channel, followed by delay-sum beamforming where the time delays of arrival are estimated using generalized cross-correlation with phase transform (GCC-PHAT) [Knapp and Carter, 1976] and smoothed using a two-stage Viterbi post-processing [Anguera et al., 2007]. The beamformed audio may then be processed in the same way as single channel speech, typically using speech activity detection (if the recording is not already segmented), followed by a speech recogniser. Hori et al. [2012] describe a system which applies a dereverberation algorithm to the multichannel audio, followed by a source separation approach comprising a speaker diarisation component based on clustered direction-of-arrival estimates, which is then used to direct a delay-sum beamformer.

More sophisticated beamforming algorithms have been proposed that take into account the correlation of the noise on different channels under spherically isotropic or cylindrically isotropic noise field assumptions. Such approaches, col-

lectively referred to as superdirective beamforming [Bitzer and Simmer, 2001], work well for speech enhancement by improving directional selectivity at lower frequencies. However, such techniques are designed for generic sounds and as such they neither take into account the unique characteristics of human speech, nor are designed specifically to improve speech recognition performance. There has been some work on designing a beamformer specifically assuming that its output will be used for speech recognition. For instance: the maximum negentropy beamformer [Kumatani et al., 2009] exploits the fact that the distribution of the subband samples of clean speech is super-Gaussian whereas the distribution of noise-corrupted speech is closer to Gaussian; LIMABEAM (likelihood maximising beamforming) [Seltzer and Stern, 2006] optimises the array processing parameters to maximise the likelihood of the recognised hypothesis given the filtered acoustic data. LIMABEAM may be thought of as explicitly optimising the beamforming to maximise speech recognition accuracy by taking acoustic model likelihood as a surrogate for accuracy. Fox and Hain [2014] extended LIMABEAM to discriminative setting and Sainath et al. [2015], also in the spirit of LIMABEAM, used a convolutional layer to mimic the filter–sum beamformer by implicitly learning steering delays and beamformer weights using CNN kernels; those were jointly optimised with the acoustic model parameters to maximise the performance of a speech recogniser.

Several researchers have explored ways to perform recognition from multiple distant microphones without performing explicit beamforming. Wölfel et al. [2006] investigated approaches in which each individual channel was separately recognised, with the recognition hypotheses combined using confusion network combination. A variant of this approach also recognises an enhanced channel obtained by beamforming, which is then added to the confusion network combination. Stolcke [2011] investigated this approach in detail on a meeting recognition task, concluding that combining the individual channels at the signal level by delay-sum beamforming is superior (in terms of both accuracy and processing time) compared to the individual channel approach. Marino and Hain [2011] performed some initial investigations training GMM-based systems on concatenated feature vectors from 2–4 microphones. This produced encouraging word error rates, similar to those obtained by beamforming the signals from the same microphones.

The work described in this chapter was performed in 2013. Since then, DSR

has again attracted much interest. The Jelinek Workshop[1] in 2015 hosted two teams working on signal– and model– level approaches to DSR; much of those efforts were focused on investigating deep learning methods. For example, Wisdom et al. [2016] proposed an architecture for source separation based on deep unfolding – an approach where an iterative generative separation model learning procedure is unfolded into a multi-layer network, and discriminatively refined. ANN were also applied to explicitly estimate beamforming parameters [Xiao et al., 2016] and multi-task trained ANNs were re-investigated for DSR with parallel acoustic data (close-talk and far-field) [Qian et al., 2016]. Zhang et al. [2016] reported substantial improvements for DSR using sequential acoustic models based on recurrent neural networks with highway connections.

Attempts to push state-of-the-art in robust and distant ASR have been also made through the REVERB[2] [Kinoshita et al., 2016] and CHiME[3] [Barker et al., 2015] evaluation campaigns, both suggesting (similarly to Hain et al. [2012]) the optimal performance of DSR systems requires multi-stage processing pipelines that involve signal enhancement, the use of microphone arrays, ANN acoustic model adaptation, cross-system adaptation and hypothesis combination, which were all incorporated in the winning system of Yoshioka et al. [2015] in the third CHiME evaluation campaign.

9.3 Learning representation from multiple channels

9.3.1 Beamforming

We use a delay-and-sum beamformer [Flanagan et al., 1985] as implemented in BeamformIt toolkit[4] by Anguera et al. [2007] as our baseline multi-channel technique. As outlined in the review section, there exist more sophisticated beamformers that can exploit certain properties of a particular working conditions such as known locations of sound sources and/or microphones and known number of microphones and their exact specification; however, in a general setting of a meeting room those are usually unknown, and delay-and-sum channel combination with an appropriate post-processing techniques offers comparable ASR

[1] http://www.clsp.jhu.edu/workshops/15-workshop/
[2] http://reverb2014.dereverberation.com/
[3] http://spandh.dcs.shef.ac.uk/chime_challenge/
[4] https://github.com/xanguera/BeamformIt

performance [Zwyssig et al., 2013].

Denote by $\mathbf{x}$ the raw waveform signal and $\mathbf{x}[n]$ a particular sample at time instance n, then weighted delay-and-sum beamformer implements the following operation:

$$\hat{\mathbf{x}}[n] = \sum_{m=1}^{M} \mathbf{W}^m[n]\mathbf{x}^m[n - \tau^{m,ref}[n]] \tag{9.1}$$

where $\mathbf{W}^m[n]$ is the weight assigned to the mth microphone (out of a total of M microphones) at a time instance n such that $\sum_{m=1}^{M} \mathbf{W}^m[n] = 1$. $\tau^{m,ref}$ denotes relative time delays of arrival (TDOA) between the mth and the reference microphones and are computed (in this work) every 250ms of speech using a generalized cross-correlation with phase transform (GCC-PHAT) [Knapp and Carter, 1976]. The reference channel in the delay-and-sum implementation of Anguera et al. [2007] is determined automatically by computing the average GCC-PHAT statistics for each pair of microphones on longer speech segments and selecting the microphone with the highest average cross-correlation. N-best GCC-PHAT TDOA estimates are then smoothed to avoid quick changes between acoustic events using the Viterbi algorithm. The weighting parameters $\mathbf{W}^m[n]$ are introduced to account for the case of non-regularly spaced or non-identical microphones, when one may want estimate their contributions independently (rather than assume a uniform scaling).

The resulting enhanced waveform signal $\hat{\mathbf{x}}$ follows the standard procedure of acoustic feature extraction, as indicated in Figure 9.1 and described in Section 3.5, the result of which is a sequence of acoustic observations denoted by $\mathbf{O}^b$.

9.3.2 Channel concatenation

The easiest way to incorporate multiple modalities into ANN is by presenting them as the additional inputs to the network. Acoustic observations from multiple channels $\bar{\mathbf{O}}_t^1 \dots \bar{\mathbf{O}}_t^m \dots \bar{\mathbf{O}}_t^M$ are concatenated and the first ANN layer implements:

$$\mathbf{h} = \phi\left(\mathbf{b} + \sum_{m=1}^{M} \mathbf{W}^{m\top}\bar{\mathbf{O}}_t^m\right) = \phi\left(\mathbf{b} + [\mathbf{W}^{1\top}, \dots \mathbf{W}^{m\top}, \dots, \mathbf{W}^{M\top}]\begin{bmatrix} \bar{\mathbf{O}}_{\mathbf{t}}^1 \\ \vdots \\ \bar{\mathbf{O}}_t^m \\ \vdots \\ \bar{\mathbf{O}}_t^M \end{bmatrix}\right) \tag{9.2}$$

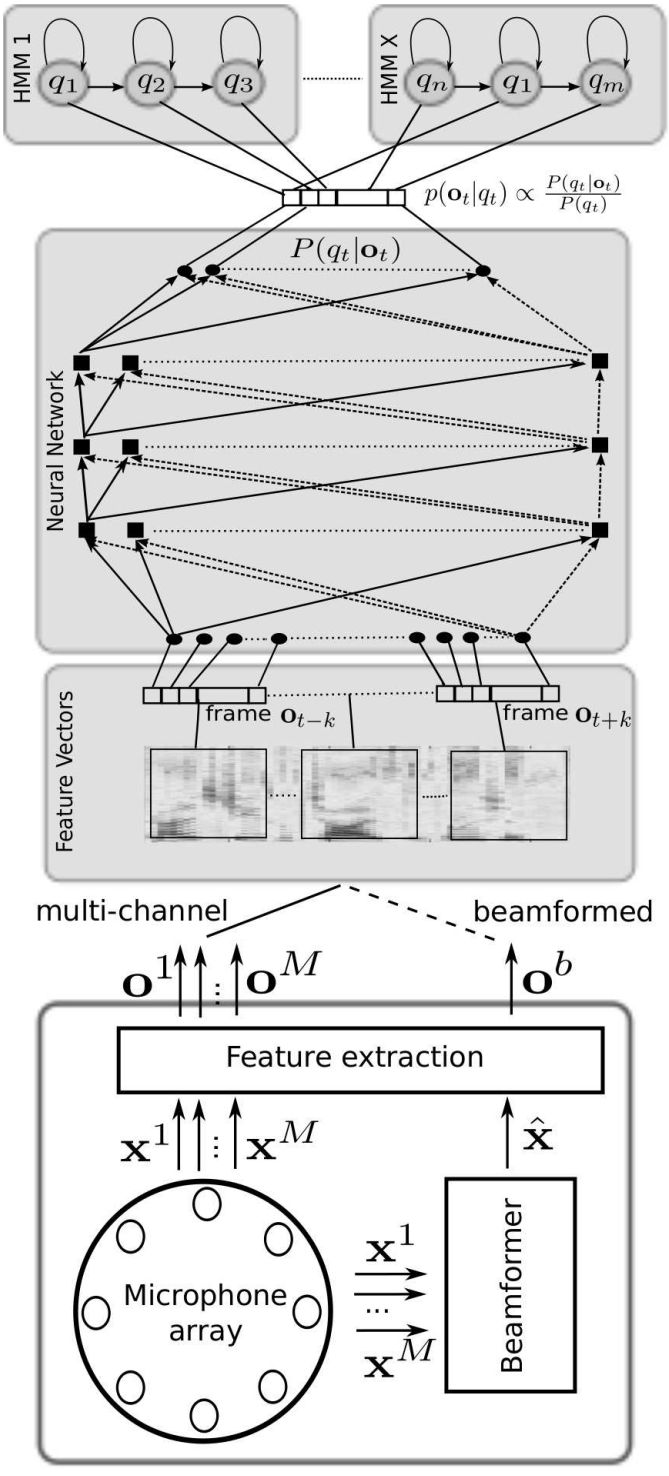


Figure 9.1: Front-end for our setups with ANN in hybrid configuration on the top.

where $\mathbf{W}^m$ is the set of parameters responsible for mth microphone. This can be modelled as a large dense matrix resulting from the concatenation of parameters responsible for particular channels, as indicated in the rightmost part of equation (9.2).

As the inputs in our scenario share similar information (acoustic observations), it is possible to reduce the number of trainable parameters and tie the weight matrices between input channels:

$$\mathbf{h} = \phi\left(\mathbf{b} + \sum_{m=1}^{M} \mathbf{W}^\top \bar{\mathbf{O}}_t^m\right). \tag{9.3}$$

This operation, similar to (9.2), can be expanded into a single dense matrix multiplication by replicating parameters $\mathbf{W}$ and concatenating the input acoustic channels as in (9.2). Tying the weights also requires an additional modification to the way the parameters are updated. The target gradient $\partial\mathcal{F}/\partial\mathbf{W}$ is expressed as the sum of the partial gradients resulting from evaluation $\mathbf{W}$ against each input microphone $\mathbf{o}_t^m$. Denote by $\mathbf{a} = \mathbf{b} + \sum_{m=1}^{M} \mathbf{W}^\top \bar{\mathbf{O}}_t^m$ the linear activations in (9.3), then the gradient is given by:

$$\frac{\partial\mathcal{F}}{\partial\mathbf{W}} = \frac{\partial\mathcal{F}}{\partial\mathbf{h}}\frac{\partial\mathbf{h}}{\partial\mathbf{a}}\frac{\partial\mathbf{a}}{\partial\mathbf{W}} \tag{9.4}$$

$$= \frac{\partial\mathcal{F}}{\partial\mathbf{h}}\frac{\partial\mathbf{h}}{\partial\mathbf{a}}\sum_{m=1}^{M}\bar{\mathbf{O}}_t^m, \tag{9.5}$$

where $\partial\mathcal{F}/\partial\mathbf{h}$ is the error signal back-propagated to the first hidden layer $\mathbf{h}$ and $\partial\mathbf{h}/\partial\mathbf{a}$ depends on a particular form of an activation function ϕ.

9.3.3 Convolutional and pooling layers

The structure of feed-forward neural networks may be enriched through the use of convolutional layers [LeCun et al., 1998b] which allow local feature receptors to be learned and reused across the whole input space. A max-pooling operator [Riesenhuber and Poggio, 1999] can be applied to downsample the convolutional output bands [Ranzato et al., 2007], thus reducing variability in the hidden activations.

9.3.3.1 Convolutional layer

Consider a neural network in which the acoustic feature vector $\mathbf{V}$ consists of re-arranged (as described later) filter-bank outputs within an acoustic context window $\bar{\mathbf{O}}_t$ of size $Z = 2c + 1$. $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_b, \ldots, \mathbf{v}_B] \in \mathbb{R}^{B \cdot Z}$ is divided

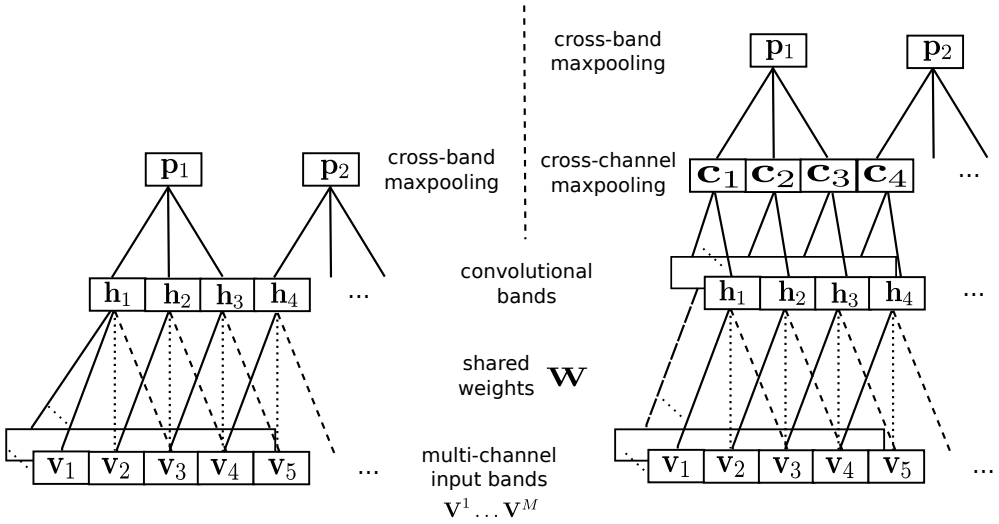


Figure 9.2: Frequency domain max-pooling multi-channel CNN layer (left), and a similar layer with cross-channel max-pooling (right). Filter weights can be either independent or shared between acoustic channels.

into B frequency bands with the b-th band $\mathbf{v}_b \in \mathbb{R}^Z$ comprising all the Z relevant coefficients (statics, Δ, Δ^2, ...) across all frames of the context window in band b. The k-th hidden convolution band $\mathbf{h}_k = [h_{1,k}, \ldots, h_{j,k}, \ldots, h_{J,k}] \in \mathbb{R}^J$ is then composed of a linear convolution of J weight vectors (filters) with F consecutive input bands $\mathbf{u}_k = [\mathbf{v}_{(k-1)L+1}, \ldots, \mathbf{v}_{(k-1)L+F}] \in \mathbb{R}^{F \cdot Z}$, where $L \in \{1, \ldots, F\}$ is the filter shift. Fig 9.2 gives an example of such a convolution with a filter size and shift of $F = 3$ and $L = 1$ respectively. This may be extended to M acoustic channels $\mathbf{V}^1 ... \mathbf{V}^M$ (each corresponding to a microphone), in which the hidden activation $h_{j,k}$ can be computed by summing over the channels:

$$h_{j,k} = \phi\left(b_{j,k} + \sum_{m=1}^{M} \mathbf{w}_j^m * \mathbf{u}_k^m\right), \tag{9.6}$$

where $\phi(\cdot)$ is a nonlinearity, $*$ denotes linear valid convolution[5], $\mathbf{w}_j^m \in \mathbb{R}^{F \cdot Z}$ is a weight vector of the j-th filter acting on the local input $\mathbf{u}_k^m$ of the m-th input channel, and $b_{j,k}$ is an additive bias for the j-th filter and k-th convolutional band. Since the channels contain similar information (acoustic features shifted in time) we conjecture that the filter weights may be shared across different channels. Nevertheless, the formulation and implementation allow for different filter weights

[5]The convolution of two vectors of size X and Y may result either in the vector of size $X + Y - 1$ for a full convolution with zero-padding of non-overlapping regions, or the vector of size $X - Y + 1$ for a valid convolution where only the points which overlap completely are considered [TheScipyCommunity, 2016].

in each channel. Similarly, it is possible for each convolutional band to have a separate learnable bias parameter instead of the biases only being shared across bands [Abdel-Hamid et al., 2012, Sainath et al., 2013a].

The complete set of convolutional layer activations $\mathbf{h} = [\mathbf{h}_1, \ldots, \mathbf{h}_K] \in \mathbb{R}^{K \cdot J}$ is composed of $K = (B - F)/L + 1$ convolutional bands obtained by applying the (shared) set of J filters across the whole (multi-channel) input space $\mathbf{V}$ (as depicted in Fig 9.2). In this work the weights are tied across the input space (i.e. each $\mathbf{u}_k$ is convolved with the same filters $\mathbf{w}_j^m$); alternatively the weights may be partially shared, tying only those weights spanning neighbouring frequency bands [Abdel-Hamid et al., 2012]. Although limited weight sharing was reported to bring improvements for phone classification [Abdel-Hamid et al., 2012] and small LVSR tasks [Abdel-Hamid et al., 2013], a recent study on larger tasks [Sainath et al., 2013a] suggests that full weight sharing with a sufficient number of filters can work equally well, while being easier to implement.

9.3.3.2 Pooling layer

A convolutional layer is usually followed by a pooling layer which downsamples the activations $\mathbf{h}$. The *max-pooling* operator [Riesenhuber and Poggio, 1999] passes forward the maximum value within a group of R activations. The s-th max-pooled band is composed of J related filters $\mathbf{p}_s = [p_{1,s}, \ldots, p_{j,s}, \ldots, p_{J,s}] \in \mathbb{R}^J$:

$$p_{j,s} = \max_{r=1}^{R} \left(h_{j,(s-1)N+r} \right), \tag{9.7}$$

where $N \in \{1, \ldots, R\}$ is a pooling shift allowing for overlap between pooling regions when $N < R$ (in Fig 9.2, $R = N = 3$). The pooling layer decreases the output dimensionality from K convolutional bands to $S = (K - R)/N + 1$ pooled bands and the resulting layer is $\mathbf{p} = [\mathbf{p}_1, ..., \mathbf{p}_S] \in \mathbb{R}^{S \cdot J}$.

9.3.3.3 Channel-wise convolution with cross-channel pooling

Multi-channel convolution (9.6) builds feature maps similarly to the LeNet-5 model [LeCun et al., 1998b] where each convolutional band is composed of filter activations spanning all input channels. We also constructed feature maps using max-pooling across channels, in which the activations $h_{j,k}^m$ are generated in channel-wise fashion and then max-pooled (9.9) to form a single cross-channel

convolutional band $\mathbf{c}_k = [c_{1,k}, \ldots, c_{j,k}, \ldots, c_{J,k}] \in \mathbb{R}^J$ (Fig 9.2 (right)):

$$h_{j,k}^m = \phi\left(b_{j,k} + \mathbf{w}_j * \mathbf{u}_k^m\right) \tag{9.8}$$

$$c_{j,k} = \max_{m=1}^{M}\left(h_{j,k}^m\right). \tag{9.9}$$

Note that here the filter weights $\mathbf{w}_j$ need to be tied across the channels such that the cross-channel max-pooling (9.9) operates on activations for the same feature receptor. The resulting cross-channel activations $\mathbf{c} = [\mathbf{c}_1, \ldots, \mathbf{c}_K] \in \mathbb{R}^{K \cdot J}$ can be further max pooled along frequency using (9.7). Channel-wise convolution may also be viewed as a special case of 2-dimensional convolution, where the effective pooling region is determined in frequency but varies in time depending on the actual time delays between the microphones.

9.4 Experiments (I)

We follow the setup described in detail in Section 4.3 using AMI data for most of the experiments (see Section 4.3.1). Some of those initial findings are further expanded to the ICSI corpus (Section 4.3.2) which provides a less constrained distant microphones setting, compared to AMI.

9.4.1 Baseline results

For AMI we use three MDM configurations where beamforming is done on 2, 4, and 8 channels respectively. The results obtained by both the BMMI-trained GMM system and the ANN systems for these configurations, as well as for the SDM and IHM conditions are shown in Table 9.1. The WERs for the GMM-based systems are comparable to the ones reported previously in [Marino and Hain, 2011, Hain et al., 2012] on AMI-based test sets, albeit using a different training-test partition.

We find the ANNs to greatly improve recognition accuracy for speech recorded with distant microphones. In fact, the network trained on SDM data is better than the best GMM-BMMI system built from beamformed audio from 8 far-field microphones. Interestingly, the ANNs are also found to be less sensitive to the number of beamformed channels used. We attribute this to the fact that multiple layers of non-linear transformations can better compensate against small variabilities in feature space [Seltzer et al., 2013, Yu et al., 2013a]. FBANK

Table 9.1: WER(%) on AMI for the GMM and ANN acoustic models for various microphone configurations.

System	Microphone configurations				
	IHM	MDM8	MDM4	MDM2	SDM
Development set (`amidev`)					
GMM BMMI on LDA/STC	30.2 (SAT)	54.8	56.5	58.0	62.3
ANN on LDA/STC	26.8 (SAT)	49.5	50.3	51.6	54.0
ANN on FBANK	26.8	49.2	-	50.1	53.1
Evaluation set (`amieval`)					
GMM BMMI on LDA/STC	31.7 (SAT)	59.4	61.2	62.9	67.2
ANN on LDA/STC	28.1 (SAT)	52.4	52.6	52.8	59.0
ANN on FBANK	29.1	52.0	-	52.4	57.9

features were also found to be better than (unadapted) LDA/STC features for ANNs and will be used in the remainder of this chapter.

While Table 9.1 presents the WER for all segments, including those with overlapped speech, Figure 9.3 shows the WERs for segments scored with different numbers of overlapped speakers. As one may expect, overlapped segments are harder to recognise. In fact, even if a beamformer can select the dominant source perfectly it still does not address the problem of recognising overlapped speech which would require source separation and independent decodes for each identified overlapping source. Figure 9.3 gives us a sense of the difficulty in recognising overlapped speech. We see an 8-12% reduction in WER when only considering segments with non-overlapping speech. One can also notice that the WERs deteriorate relatively more in the presence of overlapped speech for ANNs, for example, in the SDM case a 12% relative drop in WER is observed for the GMM-HMM and over 19% relative for the ANN-HMM system. This is expected as ANNs do in general a better job in acoustic modelling of non-overlapped segments and part of this advantage diminishes for fragments containing simultaneous speech. We do not address the issue of overlapping speakers in this chapter, and to keep the exposition simple we report WERs for all segments as they are (including overlapping speakers).

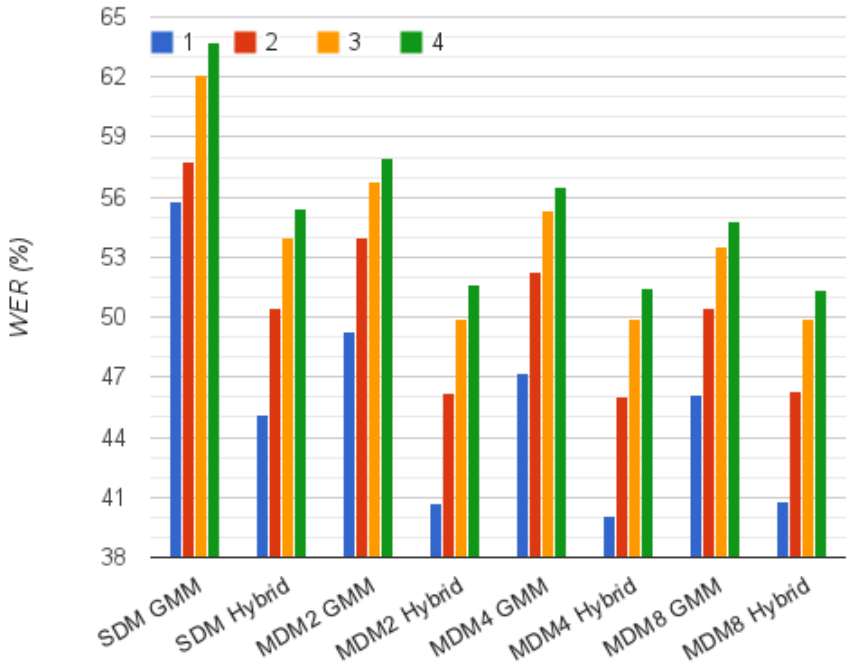


Figure 9.3: Development set WERs for segments with 1, 2, 3 and 4 overlapping speakers. AMs are trained on MFCC LDA/STC features. The Figure comes originally from [Swietojanski et al., 2013b] and the results are not directly comparable to the one reported in Table 9.1 due to the latter benefits from later refinements in the recipe. The Figure was included to visualise the overlapping speakers issue across different systems.

Table 9.2: WER for ANNs trained on multiple channels. SDM models are trained on channel 1.

Combining method	Recognition Channel(s)	amidev
SDM (no combination)	1	53.1
SDM (no combination)	2	52.9
Concatenate 1+5	3,7	51.8
Concatenate 1+3+5+7	2,4,6,8	51.7
Multi-style 1+3+5+7	1	51.8
Multi-style 1+3+5+7	2	51.7

9.4.2 Channel concatenation and data pooling

Through a second set of experiments, we evaluate the extent to which an ANN is able to learn to do front-end processing— both noise-cancellation and beamforming — by providing the features extracted from multiple microphones as input to the networks. In these initial experiments the networks again have 6 hidden layers as previously[6] except with a wider input layer. We also obtained better results in the case that the parameters were not shared between input channels, so the first layer in these experiments effectively implements equation (9.2). Note that this is not entirely comparable to the setup where the ANNs are trained on features extracted from beamformed audio, since Wiener filtering and beamforming are time domain operations, whereas the ANNs trained on concatenated features are operating entirely in cepstral or log-spectral domains. Nevertheless, the results provide us an indication of how complementary the information in different channels is. We see from Table 9.2 that the ANNs trained on concatenated inputs do in fact perform substantially better (p_v<0.001) than the SDM case, and achieve results approaching that of the beamformed configurations. The important point to note here is that the ANNs trained on concatenated features do not use any knowledge of the array geometry. Consequently the technique, similar to that of Marino and Hain [2011], is applicable to any arbitrary configuration of microphones.

To further understand the nature of the compensation being learned by the ANNs with multi-channel inputs, we performed an additional control experiment. The input to the ANN was from a single channel, and at test time this was identical to the SDM case. However, during training the data from other channels was also presented to the network, although not at the same time. In other words, the ANN is presented with data drawn from multiple channels while at test time it is only tested on a single channel. We call this the multi-style training, and it is related to our work on low-resource acoustic modelling [Ghoshal et al., 2013], where a similar concept was used to train ANNs in a multilingual fashion. From Table 9.2 we see that this approach performs similarly to the ANNs with concatenated input, without requiring multiple channels at the recognition stage. Recognition results on channel 2, which is not used in the multi-style training, show similar trends. These results strongly suggest that there is information in a single

[6]However, since the networks are being tasked with additional processing, deeper architectures may be more suitable.

Table 9.3: Word Error Rates (%) on AMI – SDM.

System	amidev
BMMI GMM-HMM (LDA/STC)	63.2
ANN (FBANK)	53.1
CNN ($R = 3$)	51.4
CNN ($R = 2$)	51.3
CNN ($R = 1$)	52.5

channel to have more accurate recognition. However, extraneous factors in the data may confound a learner trained only on data from a single channel. Being forced to classify data from multiple channels using the same shared representation (i.e. the hidden layers) the network learns how to ignore the channel-specific covariates. To the best of our knowledge, this is the first result to show that it is possible to improve recognition of audio captured with a single distant microphone by guiding the training using data from microphones at other spatial locations.

9.4.3 Convolutional Neural Networks

All CNN/ANN models in this section were trained on FBANK features appended with the first and the second time derivatives [Li et al., 2012] which were presented in symmetric context windows, with $Z = 11$ frames long.

9.4.4 SDM – Single Distant Microphone

The results of the single channel CNN can be found in Table 9.3 with the first two lines presenting the GMM and ANN baselines from Table 9.1. The following three lines are results for the CNN using max-pool sizes of $R = N = 1, 2, 3$. By using CNNs we were able to obtain 3.4% relative reduction in WER with respect to the best ANN model and a 19% relative reduction in WER compared with a discriminatively trained GMM-HMM (baseline numbers taken from Table 9.1). The total number of parameters of the CNN models varies as $R = N$ while J is kept constant across the experiments. However, the best performing model had neither the highest nor the lowest number of parameters, which suggests it is due to the optimal pooling setting.

9.4.5 MDM – Multiple Distant Microphones

For the MDM case we compared the models trained on FBANK features extracted from a signal enhanced by delay-sum beamformer with the use of FBANK features extracted from multiple microphone channels and presented directly as an expanded input to the network (see also Fig. 9.1). For the beamforming experiments, we follow noise cancellation using a Wiener filter with delay-sum beamforming on 8 uniformly-spaced array channels using the BeamformIt toolkit [Anguera et al., 2007][7]. The results are summarised in Table 9.4. The first block of Table 9.4 presents the results for the case in which the models were trained on a beamformed signal from 8 microphones. The first two rows show the WER for the baseline GMM and ANN acoustic models as reported in Table 9.1. The following row contains the CNN model trained on 8 beamformed channels obtaining 2.7% absolute improvement (5.5% relative) over ANN. The configuration of the MDM CNN is the same as the best SDM CNN ($R = N = 2$).

The second part of a Table 9.4 shows WERs for the models directly utilising multi-channel features. The first row is a baseline ANN variant trained on 4 concatenated channels from Table 9.2. Then we present the CNN models with MDM input convolution performed as in equation (9.6) and pooling size of 2, which was optimal for the SDM experiments. This scenario decreases WER by 1.6% relative when compared to the ANN structure with concatenated channels (this approach can be seen as a channel concatenation for CNN models). Applying channel-wise convolution with two-way pooling (outlined in section 9.3.3.3) brings further gains of 3.5% WER relative. Furthermore, channel-wise pooling works better for more input channels: conventional convolution on 4 channels achieves 50.4% WER, practically the same as the 2 channel network, while channel-wise convolution with 4 channels achieves 49.5% WER, compared to 50.0% for the 2-channel case. These results indicate that picking the best information (selecting the feature receptors with maximum activations) within the channels is crucial when doing model-based combination of multiple microphones.

9.4.5.1 Different weight-sharing techniques

When using multiple distant microphones directly as the input to a CNN, we posit that the same filters should be used across the different channels even when

[7] We followed noise-cancellation pipeline in our earlier recipe. The one currently available in the Kaldi repository does not perform Wiener-filtering by default.

Table 9.4: Word Error Rates (%) on AMI – MDM.

System	amidev
MDM with beamforming (8 microphones)	
BMMI GMM-HMM	54.8
ANN	49.2
CNN	46.8
MDM without beamforming	
ANN 4ch concatenated	51.2
CNN 2ch conventional	50.5
CNN 4ch conventional	50.4
CNN 2ch channel-wise	50.0
CNN 4ch channel-wise	49.4

Table 9.5: Word Error Rates (%) on AMI MDM without beamformer.

System	amidev
CNN 2ch tied $\mathbf{w}_j^m$	50.5
CNN 2ch not tied $\mathbf{w}_j^m$	51.3

cross-channel pooling is not used. Each channel contains the same information, albeit shifted in time, hence using the same feature detectors for each channel is a prudent constraint to learning. This is confirmed in Table 9.5 where a non-tied variant with a separate set of filters for each channel is 0.7% absolute worse (p_{v}<0.001) compared to the case when the filter weights are shared across channels.

9.5 Experiments (II)

Based on our previous findings in Section 9.4 we extend the study to the ICSI data (Section 4.3.2) and to ANN/CNN models with different non-linearities – sigmoid, ReLU and Maxout. ReLU models have the same structure as sigmoid ones (2k hidden units, 6 hidden layers) and maxout networks were tuned to have a similar number of parameters with six hidden layers, resulting in 1150 maxout units and pool size $R = 3$. Convolutional layers in all models, similarly to previous experiments, were configured to have $J = 128$ filters.

For the ReLU and Maxout models we sample initial weights from a uniform

Table 9.6: WER (%) on AMI and ICSI – SDM.

System	amidev	icsieval
BMMI GMM-HMM (LDA/STC)	63.2	56.1
ANN– Sigmoid	53.1	47.8
ANN– ReLU	51.1	46.3
ANN- Maxout	50.8	45.9
CNN – Sigmoid	51.3	46.5
CNN – ReLU	50.3	45.6
CNN – Maxout	50.5	45.6

distribution with range $(-0.005, 0.005)$ (see Section 4.2 for sigmoid configuration). All models are finetuned with the exponentially decaying `newbob` learning rate schedule staring from an initial learning rate of 0.08 (for sigmoid, Section 4.2) and 0.01 for piece-wise linear activations.

9.5.1 SDM – Single Distant Microphone

The SDM experiments used the first microphone from the AMI circular array and the second tabletop boundary microphone from the ICSI recordings. Our results are shown in Table 9.6, with the three baseline systems in line 1 (BMMI GMM), line 2 (ANN– Sigmoid), and line 5 (CNN-Sigmoid). The ANN baseline has a 15% relative lower WER than the discriminative GMM baseline, with the CNN baseline improving over the ANN baseline by a further 3% relative. Comparing the ReLU and Maxout ANN and CNN systems, with the sigmoid baselines, shows a consistent improvement in WER of 1.5–4.5%. Comparing ANNs and CNNs with the same activation function, we see that networks with the sigmoid nonlinearity benefit the most from a convolutional layer (3–4% relative reduction in WER), although the ReLU and Maxout systems do benefit from the use of a convolutional layer (0.5–2% relative). We note that these experiments have been performed with a fixed number of filters, optimised for sigmoid-based systems; further experiments are needed to ascertain if the ReLU and Maxout systems would give large decreases in WER if there were more convolutional filters.

Table 9.7: WER (%) on AMI and ICSI – MDM8 with beamforming

System	amidev	icsieval
BMMI GMM-HMM (LDA/STC)	54.8	46.8
ANN– Sigmoid	49.2	41.0
ANN– ReLU	46.3	38.7
ANN– Maxout	46.4	39.0
CNN – Sigmoid	46.3	39.5
CNN – ReLU	46.0	37.6
CNN – Maxout	45.9	38.1

Table 9.8: WER (%) on AMI and ICSI – MDM4 with multi-channel input

System	amidev	icsieval
CNN – Sigmoid (conventional)	50.4	43.3
CNN – Sigmoid (channel-wise)	49.5	40.1
CNN – ReLU (channel-wise)	48.7	37.5
CNN – Maxout (channel-wise)	48.4	37.8

9.5.2 MDM – Multiple Distant Microphones

For the MDM systems we consider: (1) *beamforming* the signal into a single channel (using all 8 microphones for AMI and 4 tabletop boundary microphones for ICSI) and following the standard acoustic modelling approaches used for the SDM case in Section 9.4; (2) *cross-channel pooling* using a channel-wise convolutional layer for training on 4 microphone channels, in which the hidden activations are constructed from the maximum activations across the channels. The ICSI data is characterised by large distances between microphones, and picking the right microphone for a talker is crucial, which may be well-matched to cross-channel pooling.

Table 9.7 shows the results for the models trained on a single beamformed channel (following the procedure described in Section 9.4) We observe similar reductions in WER for sigmoid CNNs over ANNs as in the SDM case. The gain of CNN variants using ReLUs and Maxout in place of sigmoid activation functions remains small. These trends can be observed for both the AMI and ICSI datasets. We note that the WERs obtained using the ANN or CNN models (table 9.6) are lower than the WERs obtained for the discriminative GMM systems in the MDM

Table 9.9: Word Error Rates (%) on AMI – IHM

System	amidev
BMMI GMM-HMM (LDA/STC, SAT)	30.2
ANN– Sigmoid	26.6
ANN– ReLU	25.5
ANN– Maxout	26.3
CNN – Sigmoid	25.6
CNN – ReLU	24.9
CNN – Maxout	25.0

case trained on a beamformed signal.

Table 9.8 shows the results obtained for CNNs trained using multi-channel input without beamforming. The first row presents a "conventional" approach where convolutional activations are produced by a sum of filter activations from each channel. Since that was found to be especially harmful for less constrained microphone configurations (ICSI) the following rows present a channel-wise approach where only the maximum activations within the channels are considered using cross-channel pooling described in Section 9.3.3.3. For the AMI data the CNN architectures return similar WERs to ANNs using beamformed input; for the ICSI data CNNs using cross-channel pooling match the WERs obtained using beamforming, which we hypothesise is due to less accurate TDOA estimates from the uncalibrated microphone array.

9.5.3 IHM – Individual Headset Microphone

For comparison purposes we present WERs for the different architectures using close-talking IHM inputs, for the AMI data (Table 9.9). The WER trend is similar to the distant microphone cases, suggesting that the results for the different nonlinear activations generalise across signal qualities. BMMI-GMM models were estimated using speaker adaptive training with fMLLR.

9.6 Summary and Discussion

We have investigated using CNNs for DSR with single and multiple microphones. A CNN trained on a single distant microphone is found to produce a WER

approaching these of a ANN trained using beamforming across 8 microphones. In experiments with multiple microphones, we compared CNNs trained on the output of a delay-sum beamformer with those trained directly on the outputs of multiple microphones. In the latter configuration, channel-wise convolution followed by a cross-channel max-pooling was found to perform better than multi-channel convolution.

A beamformer uses time-delays between microphone pairs whose computation requires knowledge of the microphone array geometry, while these convolutional approaches need no such knowledge. CNNs are able to compensate better for the confounding factors in distant speech than ANNs. However, the compensation learned by CNNs is complementary to that provided by a beamformer. In fact, when using CNNs with cross-channel pooling, similar WERs were obtained by changing the order of the channels at test time from the order in which they were presented at training time, suggesting that the model is able to pick the most informative channel. This idea has been further extended recently by Kim and Lane [2015] to cross-channel pooling using attention mechanism.

Early work on CNNs for ASR focussed on learning shift-invariance in time [Waibel et al., 1989, Lee et al., 2009], while more recent work [Abdel-Hamid et al., 2012, Sainath et al., 2013a] have indicated that shift-invariance in frequency is more important for ASR. The results presented here suggest that recognition of distant multichannel speech is a scenario where shift-invariance in time between channels is also important, thus benefitting from pooling in both time and frequency.

The presented distant conversational speech recognition experiments have explored a number of different neural network architectures, using different nonlinear functions for the hidden layer activations. Our results, using the AMI and ICSI corpora, show that neural network acoustic models offer large reductions in WER compared with discriminatively trained GMM-based systems on DSR. Furthermore, we observed further significant reductions in WER by using a convolutional layer within a ANN architecture. Small, but consistent, reductions in WER were also obtained by using ReLU and Maxout activation functions in place of sigmoids.

These neural network based systems used log spectral input representations, which are potentially amenable to additional feature space transformations and modelling. In particular, our current experiments do not explicitly attempt to

optimise the acoustic model for overlapping talkers, or for reverberation. The promising results using raw multiple channel input features in place of beamforming opens the possibilities to learning representations taking into account aspects such as overlapping speech.

Chapter 10

Conclusions

10.1 Overview of contributions

This thesis addressed three aspects of representation learning using ANNs for acoustic modelling in hybrid systems. Below we summarise those findings from the contribution parts:

- **Part II – Adaptation**: We have developed and investigated two techniques for unsupervised ANN speaker and environment adaptation. The first technique, termed learning hidden unit contributions (`LHUC`) and described in Chapter 5, operates in model-space and performs adaptation by learning new combination coefficients for a speaker-independent (SI) basis in a speaker-dependent (SD) manner. Because the SI basis may not be optimal for unseen data, we propose a speaker adaptive trained `LHUC` (`SAT-LHUC`) which retains the information necessary to model the individual characteristics of the speakers in training data (not just their average aspect) and thus offers as a result more tunable canonical representation. `(SAT-)LHUC` operates at the level of a single hidden unit and does not allow the units to be additionally recombined with each other in a SD manner for which reason we proposed to carry the adaptation with parametric and differentiable pooling operators in Chapter 6. More specifically, we have investigated two such parameterisations based on L_p-norm (`Diff-`L_p) and Gaussian (`Diff-Gauss`) kernels inserted in each hidden ANN layer. Parameters of such ANNs are trained in standard way using error-backpropagation and pooling parameters are later altered in a SD manner in the adaptation stage. We evaluated `(SAT-)LHUC`, `Diff-`L_p and `Diff-Gauss` techniques us-

ing three benchmark corpora allowing to simulate different aspects of adaptation, in particular, the amount of adaptation data per speaker, the impact of quality of both data and the associated adaptation targets, complementarity to other adaptation techniques and (for `LHUC`) adapting ANN models trained in sequence discriminative manner and factorisation of acoustic environments. We found the proposed techniques to improve ASR performance. On average, after `LHUC` and `SAT-LHUC` adaptation to 200 speakers of TED, AMI and Switchboard data relative WER reductions of 7.0% and 9.7% were observed with respect to SI models. Differentiable pooling versions were found to work better in the SI case and the gains from adaptation were comparable in relative terms to the ones obtained with `LHUC`, which was also to be found complementary in the joint (`LHUC` + `Diff`) setting.

- **Part III – Low-resource acoustic modelling**: We focused primarily on the challenge of building ASR systems in under-resourced conditions focusing on insufficient amounts of transcribed acoustic material for estimating acoustic models in the target language – thus assuming resources like lexicons or texts to estimate language models were available. Chapter 7 was dedicated to approaches designed to work primarily with the target language resources. We proposed an ANN with a structured output layer (`SOL-ANN`) where the output layer comprises two tasks responsible for modelling context–dependent (CD) and context–independent (CI) speech units, and the CI predictions are used at runtime to aid the prediction of CD states. This approach, that can be considered as a form of curriculum learning, leads to consistent gains in SI low-resource acoustic modelling (on average 6.4% and 4.4% relative WER decrease for scenarios of 10 and 30 hours of TED lectures training material, respectively). As adaptation is an inherently low-resource problem, we also proposed to use `SOL-ANN` to multi-task adaptation in which a CD model is adapted using an auxiliary layer of CI targets which, given the same amount of adaptation material, are characterised by lower sparsity. The advantage of multi-task `LHUC` adaptation is again only visible for low-resource scenarios bringing a relative average 11.7% and 13.6% reduction in WER for 10 and 30 hours training scenarios, respectively. For non multi-task `LHUC` adaptation the same numbers were 10.7% and 12.6% for 10 and 30 training conditions, respectively.

Chapter 8 investigated the possibility of unsupervised multi-lingual knowledge transfer. Here we assume we have unconstrained access to untranscribed audio material, possibly in many source languages, and our goal is to use it to aid the estimation of acoustic model for the target language. To do so, we initialise our target acoustic model with stacked Gaussian-Bernoulli and Bernoulli-Bernoulli restricted Boltzmann machines (RBMs) using multi-lingual acoustics in an unsupervised manner. We found that pre-training seems to be language–independent, and initialisation with anyone of the considered source languages yielded similar acoustic models in terms of the obtained accuracies as when untranscribed acoustics only from the target language were used (differences are statistically insignificant). On the other hand, pre-trained models consistently obtained significantly lower WERs when compared to un-pretrained models by 6.7%, 9.0% and 14.4% for 15, 5 and 1 hours of training data, respectively (on the GlobalPhone German evaluation set).

- **Part IV – Distant speech recognition**: Our main focus within this avenue of work was to incorporate multiple channels of acoustic information into the model. In Chapter 9 we compared signal-level beamforming of multiple microphones with channel concatenation in an ANN framework. Channel concatenation gives little control on how the information is combined, especially when microphones are spaced in large distances from each other. This motivated the use of convolutional neural networks (CNN) enriched with a two–level pooling mechanism, one reducing variability across frequency and the other one working along different microphones, constructing an intermediate representation where each convolutional band (which is related to the frequency region of an acoustic observation) is independently selected in a channel-wise manner. We evaluated the proposed models on AMI and ICSI meetings data, each characterised by different configuration of distant microphones, and found our proposition of CNN with cross-channel pooling improves the accuracies when compared with channel concatenation, by allowing it to make an efficient use of higher number of microphones. For less constrained microphones arrays (ICSI) our multi-channel CNN models with cross channel pooling were able match comparable CNN models estimated from acoustic features that were extracted from

a signal enhanced by a beamformer (that operated on the same number of distant microphones).

10.2 Further work

- **Adaptation:** Our investigations concerned only feed-forward (convolutional) ANN models and it would be interesting to explore which and to what extent our propositions and findings are applicable to other ANN structures, with recurrent and time-delay ANNs being good candidates. For the time-delay variant of ANN we expect our findings on `(SAT-)LHUC` and differentiable pooling operators to be fully transferable. The use of `LHUC` with recurrent models would require more careful treatment as the default use by simple multiplication of each hidden unit by a fixed SD scale would lead exponentially to saturation (assuming a squashing non-linearity). The cell gate in the long-short term memory variant of RNN (LSTM) can, in fact, be seen as a form of data-dependent variable `LHUC` which determines the output of the model at each time step. This, in connection with the successful use of `LHUC` in Chapter 5, suggests that speaker adaptation of LSTM models could be carried out by altering, or learning in a speaker-adaptive manner, the parametrisation of an input gate. In contrast, `Diff-`L_p and `Diff-Gauss` do not possess the aforementioned `LHUC` characteristic (the gain is normalised within a pool) and their application to train and adapt RNN models would be an interesting experiment to perform. While the RNN direction is definitely interesting, we have not fully covered some relevant aspects of the proposed techniques. For example, it is to be seen to what extent adaptation with differentiable pooling operators will benefit acoustic models trained in a sequence discriminative manner, how to extend it to speaker adaptive training or how to use it for environment factorisation.

- **Low-resource acoustic modelling:** This aspect, to date, has been reasonably well investigated in the context of estimating acoustic models from small amounts of data jointly with other less-data-constrained tasks. At the same time, most of techniques related to construction of ASR systems require some starting resources to bootstrap initial seed systems, which can be then iteratively improved in (semi–)supervised manner. An interest-

ing direction, at least from an academic standpoint, is the zero-resource ASR problem, and it would be interesting to see whether cross-lingual pre-training (based on either stacked RBMs or auto-encoders) can aid some stages of bootstrapping seed models in a purely unsupervised setting.

- **Distant speech recognition:** As we have mentioned in the review part of Chapter 9, overlapping speech is a largely unsolved challenge which requires special treatment and parallel processing by tracking (usually using a beamformer) relevant sound sources and enhancing their signal with respect to competing acoustic sources. The result of this stage are multiple parallel acoustic channels which produce independent recognition hypotheses. The CNN layer with cross-channel pooling would be a good base for attempts to perform such a separation in acoustic model-space (though at the current stage the model does not address the necessity of parallel decodes). This approach would be well suited to scenarios where speakers are dominant in different microphones and in cases where their speech occupies different frequency regions (for example, male and female voices) – the latter is related to spectral masking in feature space, except here would be performed at model-level and the CNN filters could be deliberately tuned for this purpose to maximise ASR performance.

  Beamforming heavily relies on phase information when performing microphone combination while our approach only depends on amplitude of the signal – and as observed, works particularly well in scenarios where the difference between microphones is significant; in already mentioned work on multi-microphone attention pooling adding phase information as an auxiliary feature greatly extended ASR performance. We believe the ultimate solution for incorporating multiple microphones will rely on complex-valued neural networks. One could treat a convolutional layer as implementing a Fourier transform and CNN parameters comprises both real and imaginary components. This would allow to explicitly capture phase information between input channels, and use it later to aid model-level beamforming.

Bibliography

O Abdel-Hamid and H Jiang. Fast speaker adaptation of hybrid NN/HMM model for speech recognition based on discriminative learning of speaker code. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, pages 4277–4280, 2013.

O Abdel-Hamid and H Jiang. Rapid and effective speaker adaptation of convolutional neural network based models for speech recognition. In *Proc. ISCA Interspeech*, pages 1248–1252, 2013.

O Abdel-Hamid, A-R Mohamed, J Hui, and G Penn. Applying convolutional neural networks concepts to hybrid NN–HMM model for speech recognition. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, pages 4277–4280, 2012.

O Abdel-Hamid, L Deng, and D Yu. Exploring convolutional neural network structures and optimization techniques for speech recognition. In *Proc. ISCA Interspeech*, 2013.

O Abdel-Hamid, AR 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, 2014. ISSN 2329-9290. doi: 10.1109/TASLP.2014.2339736.

V Abrash, H Franco, A Sankar, and M Cohen. Connectionist speaker normalization and adaptation. In *Proc. Eurospeech*, pages 2183–2186, 1995.

JE Adcock, Y Gotoh, DJ Mashao, and HF Silverman. Microphone-array speech recognition via incremental MAP training. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, pages 897–900, 1996.

T Anastasakos, J McDonough, R Schwartz, and J Makhoul. A compact model for speaker-adaptive training. In *Proc. Int. Conf. Spoken Language Processing (ICSLP)*, pages 1137–1140, 1996.

X Anguera, C Wooters, and J Hernando. Acoustic beamforming for speaker diarization of meetings. *IEEE Trans. Audio, Speech, & Lang. Process.*, 15: 2011–2021, 2007.

G Aradilla, H Bourlard, and M Magimai-Doss. Using KL-based acoustic models in a large vocabulary recognition task. Technical Report Idiap-RR-14-2008, IDIAP, 2008.

M Bacchiani and D Rybach. Context dependent state tying for speech recognition using deep neural network acoustic models. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, pages 230–234, 2014.

L Bahl, P Brown, P de Souza, and R Mercer. Maximum mutual information estimation of hidden markov model parameters for speech recognition. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, volume 11, pages 49–52, 1986. doi: 10.1109/ICASSP.1986.1169179.

JK Baker. *Stochastic modeling as a means of automatic speech recognition*. PhD thesis, Carnegie Mellon University, 1975.

J Barker, R Marxer, E Vincent, and S Watanabe. The third CHiME speech separation and recognition challenge: Dataset, task and baselines. In *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2015.

AR Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3):930–945, 1993.

LE Baum and JA Eagon. An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. *Bull. Amer. Math. Soc.*, 73(3):360–363, 05 1967. URL `http://projecteuclid.org/euclid.bams/1183528841`.

P Bell. *Full Covariance Modelling for Speech Recognition*. PhD thesis, University of Edinburgh, 2010.

P Bell and S Renals. Regularization of context-dependent deep neural networks with context-independent multi-task training. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, 2015.

P Bell, P Swietojanski, and S Renals. Multi-level adaptive networks in tandem and hybrid ASR systems. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, 2013.

P Bell, P Swietojanski, J Driesen, M Sinclair, F McInnes, and S Renals. The UEDIN system for the IWSLT 2014 evaluation. In *Proc. International Workshop on Spoken Language Translation (IWSLT)*, pages 26–33, 2014.

P Bell, P Swietojanski, and S Renals. Multitask learning of context-dependent targets in deep neural network acoustic models. *Submitted to IEEE/ACM Transactions on Audio, Speech, and Language Processing*, -(-):1–1, 2016.

Y Bengio, R Ducharme, P Vincent, and Ch Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155, March 2003. ISSN 1532-4435. URL `http://dl.acm.org/citation.cfm?id=944919.944966`.

Y Bengio, P Lamblin, D Popovici, and H Larochelle. Greedy layer-wise training of deep networks. In *Advances in Neural Information Processing Systems 19*, pages 153–160. 2007.

Y Bengio, J Louradour, R Collobert, and J Weston. Curriculum learning. In *Proc. Int. Conf. Machine Learning (ICML)*, 2009.

Y Bengio, A Courville, and P Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013. ISSN 0162-8828. doi: http://doi.ieeecomputersociety.org/10.1109/TPAMI.2013.50.

J Bilmes. Graphical models and automatic speech recognition. In *Mathematical Foundations of Speech and Language Processing*. Springer-Verlag, 2003.

J Bilmes and C Bartels. Graphical model architectures for speech recognition. *IEEE Signal Processing Magazine*, 22(5):89–100, September 2005.

CM Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

J Bitzer and KU Simmer. Superdirective microphone arrays. In M Brandstein and D Ward, editors, *Microphone Arrays*, pages 19–38. Springer, 2001.

Y-L Boureau, J Ponce, and Y LeCun. A theoretical analysis of feature pooling in visual recognition. In *Proc. Int. Conf. Machine Learning (ICML)*, 2010.

H Bourlard and N Morgan. A continuous speech recognition system embedding MLP into HMM. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, pages 186–193. 1990.

H Bourlard and N Morgan. *Connectionist Speech Recognition: A Hybrid Approach*. Kluwer Academic Publishers, 1994.

H Bourlard and CJ Wellekens. Links between Markov models and multilayer perceptrons. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(12):1167–1178, 1990.

H Bourlard, N Morgan, C Wooters, and S Renals. CDNN: a context dependent neural network for continuous speech recognition. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, volume 2, pages 349–352, 1992. doi: 10.1109/ICASSP.1992.226048.

JS Bridle. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In F Fogelman Soulié and J Hérault, editors, *Neurocomputing*, pages 227–236. Springer, 1990.

JS Bridle and S Cox. Recnorm: Simultaneous normalisation and classification applied to speech recognition. In *Proc. Advances in Neural Information and Processing Systems (NIPS)*, 1990.

PF Brown, PV deSouza, RL Mercer, VJD Pietra, and JC Lai. Class-based n-gram models of natural language. *Comput. Linguist.*, 18(4):467–479, December 1992. ISSN 0891-2017.

U Bub, J Kohler, and B Imperl. In-service adaptation of multilingual hidden-markov-models. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, volume 2, pages 1451–1454, 1997. doi: 10.1109/ICASSP.1997.596222.

L Burget, P Schwarz, M Agarwal, P Akyazi, K Feng, A Ghoshal, O Glembek, N Goel, M Karafiát, and D Povey. Multilingual acoustic modeling for speech recognition based on subspace gaussian mixture models. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, pages 4334–4337, 2010.

M Cai, Y Shi, and J Liu. Deep maxout neural networks for speech recognition. In *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 291–296, 2013.

J Carletta. Unleashing the killer corpus: experiences in creating the multi-everything AMI Meeting Corpus. *Language Resources & Evaluation*, 41:181–190, 2007.

MÁ Carreira-Perpiñán and W Wang. Distributed optimization of deeply nested systems. In *Proc. Artificial Intelligence and Statistics Conf. (AISTATS)*, pages 10–19, 2014.

R Caruana. Multitask learning. *Machine learning*, 28:41–75, 1997.

Ö Çetin, M. Magimai.-Doss, K. Livescu, A. Kantor, S. King, C. Bartels, and J. Frankel. Monolingual and crosslingual comparison of tandem features derived from articulatory and phone MLPs. In *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2007.

M Cettolo, C Girardi, and M Federico. Wit^3: Web inventory of transcribed and translated talks. In *Proc. European Association for Machine Translation (EAMT)*, pages 261–268, 2012.

M Cettolo, C Girardi, and M Federico. Report on the 10th IWSLT evaluation campaigns. In *Proc. International Workshop on Spoken Language Translation (IWSLT)*, 2013.

W Chan, N Jaitly, QV Le, and O Vinyals. Listen, Attend and Spell. *ArXiv e-prints*, 2015.

D Chen, B Mak, C-C Leung, and S Sivadas. Joint acoustic modelling of triphones and trigraphemes by multi-task learning deep neural networks for low-resource speech recognition. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, 2014.

J Cooley and J Tukey. An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, 19(90):297–301, 1965.

G Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

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

FJ Damerau. *Markov models and linguistic theory: an experimental study of a model for English*. Janua linguarum: Series minor. Mouton, 1971. URL `https://books.google.co.uk/books?id=qxBZAAAAMAAJ`.

KH Davis, R Biddulph, and S Balashek. Automatic recognition of spoken digits. *The Journal of the Acoustical Society of America*, 24(6):637–642, 1952.

J Dean, G Corrado, R Monga, K Chen, M Devin, M Mao, A Senior, P Tucker, K Yang, and QV Le. Large scale distributed deep networks. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, pages 1223–1231, 2012.

N Dehak, PJ Kenny, R Dehak, P Dumouchel, and P Ouellet. Front end factor analysis for speaker verification. *IEEE Trans Audio, Speech and Language Processing*, 19:788–798, 2010.

M Delcroix, K Kinoshita, T Hori, and T Nakatani. Context adaptive deep neural networks for fast acoustic model adaptation. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, 2015.

AP Dempster, NM Laird, and DB Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of The Royal Statistical Society, Series B*, 39(1):1–38, 1977.

D Erhan, Y Bengio, A Courville, P-A Manzagol, and P Vincent. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11:625–660, 2010.

M Federico, M Cettolo, L Bentivogli, M Paul, and S Stüker. Overview of the IWSLT 2012 evaluation campaign. In *Proc. International Workshop on Spoken Language Translation (IWSLT)*, 2012.

J Fiscus, WM Fisher, AF Martin, MA Przybocki, and DS Pallett. 2000 NIST evaluation of conversational speech recognition over the telephone: English and Mandarin performance results. In *Proc. Speech Transcription Workshop*. Citeseer, 2000.

JG Fiscus, J Ajot, N Radde, and C Laprun. Multiple dimension Levenshtein edit distance calculations for evaluating ASR systems during simultaneous speech. In *Proc. Language Resources and Evaluation Conference (LREC)*, 2006.

E Fix and JL Hodges. Discriminatory analysis, nonparametric discrimination: Consistency properties. *US Air Force School of Aviation Medicine*, Technical Report 4(3):477+, January 1951.

JL Flanagan, JD Johnston, R Zahn, and GW Elko. Computer-steered microphone arrays for sound transduction in large rooms. *The Journal of the Acoustical Society of America*, 78(5):1508–1518, 1985.

GD Forney. The Viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, March 1973. ISSN 0018-9219. doi: 10.1109/PROC.1973.9030.

C Fox and T Hain. Extending Limabeam with discrimination and coarse gradients. In *Proc. ISCA Interspeech*, pages 2440–2444, 2014.

H Franco, M Cohen, N Morgan, D Rumelhart, and V Abrash. Context-dependent connectionist probability estimation in a hybrid HMM-neural net speech recognition system. *Computer Speech and Language*, 8:211–222, 1994.

RM French. Catastrophic forgetting in connectionist networks: Causes, consequences and solutions. *Trends in Cognitive Sciences*, 3:128–135, 1999. URL `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.36.3676`.

K Fukushima and S Miyake. Neocognitron: A new algoriothm for pattern recognition tolerant of deformations. *Pattern Recognition*, 15:455–469, 1982.

MJF Gales. Maximum likelihood linear transformations for HMM-based speech recognition. *Computer Speech and Language*, 12:75–98, April 1998.

MJF Gales. Semi-tied covariance matrices for hidden Markov models. *IEEE Transactions on Speech and Audio Processing*, 7(3):272–281, 1999.

MJF Gales. Cluster adaptive training of hidden Markov models. *IEEE Transactions on Speech and Audio Processing*, 8(4):417–428, 2000.

MJF Gales and S Young. The application of hidden Markov models in speech recognition. *Foundations and trends in signal processing*, 1(3):195–304, 2008.

J Gehring, Y Miao, Fn Metze, and A Waibel. Extracting deep bottleneck features using stacked auto-encoders. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, pages 3377–3381. IEEE, 2013.

A Ghoshal, P Swietojanski, and S Renals. Multilingual training of deep neural networks. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, 2013. doi: 10.1109/ICASSP.2013.6639084.

D Gillick, L Gillick, and S Wegmann. Don't multiply lightly: Quantifying problems with the acoustic model assumptions in speech recognition. In *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 71–76. IEEE, 2011.

L Gillick and SJ Cox. Some statistical issues in the comparison of speech recognition algorithms. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, pages 532–535 vol.1, May 1989. doi: 10.1109/ICASSP.1989.266481.

JJ Godfrey, EC Holliman, and J McDaniel. Switchboard: Telephone speech corpus for research and development. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, pages 517–520. IEEE, 1992.

IJ Goodfellow, D Warde-Farley, M Mirza, A Courville, and Y Bengio. Maxout networks. In *Proc. Int. Conf. Machine Learning (ICML)*, pages 1319–1327, 2013.

RA Gopinath. Maximum likelihood modeling with Gaussian distributions for classification. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, volume 2, pages 661–664, May 1998. doi: 10.1109/ICASSP.1998.675351.

A Graves and N Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *Proc. Int. Conf. Machine Learning (ICML)*, pages 1764–1772, 2014.

F Grezl, M Karafiat, S Kontar, and J Cernocky. Probabilistic and bottle-neck features for LVCSR of meetings. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, pages IV–757–IV–760, 2007.

F Grézl, M Karafiát, and M Janda. Study of probabilistic and bottle-neck features in multilingual environment. In *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2011.

C Gülçehre, K Cho, R Pascanu, and Y Bengio. Learned-norm pooling for deep feedforward and recurrent neural networks. In *Proc. Eur. Conf. Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD)*, pages 530–546. Springer-Verlag, 2014.

A Gunawardana, M Mahajan, A Acero, and JC Platt. Hidden conditional random fields for phone classification. In *Proc. ISCA Interspeech*, pages 1117–1120, 2005.

V Gupta, P Kenny, P Ouellet, and T Stafylakis. I-vector based speaker adaptation of deep neural networks for French broadcast audio transcription. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, 2014.

R Haeb-Umbach and H Ney. Linear discriminant analysis for improved large vocabulary continuous speech recognition. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, pages 13–16, 1992. ISBN 0-7803-0532-9. URL `http://dl.acm.org/citation.cfm?id=1895550.1895555`.

T Hain, L Burget, J Dines, PN Garner, F Grézl, A El Hannani, M Karafíat, M Lincoln, and V Wan. Transcribing meetings with the AMIDA systems. *IEEE Transactions on Audio, Speech and Language Processing*, 20:486–498, 2012.

G Heigold, V Vanhoucke, A Senior, P Nguyen, M Ranzato, M Devin, and J Dean. Multilingual acoustic models using distributed deep neural networks. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, 2013.

J Hennebert, C Ris, H Bourlard, S Renals, and N Morgan. Estimation of global posteriors and forward-backward training of hybrid HMM/ANN systems. In *Proc. Eurospeech*, pages 1951–1954, Rhodes, 1997.

GE Henter, T Merritt, M Shannon, C Mayo, and S King. Measuring the perceptual effects of modelling assumptions in speech synthesis using stimuli constructed from repeated natural speech. In *Proc. ISCA Interspeech*, volume 15, pages 1504–1508, September 2014.

H Hermansky. Perceptual linear predictive (PLP) analysis of speech. *The Journal of the Acoustical Society of America*, 87(4):1738–1752, April 1990.

H Hermansky, DPW Ellis, and S Sharma. Tandem connectionist feature extraction for conventional HMM systems. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, pages 1635–1638, 2000.

I Himawan, P Motlicek, M Ferras, and S Madikeri. Towards utterance-based neural network adaptation in acoustic modeling. In *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2015.

G Hinton, L Deng, D Yu, GE Dahl, A Mohamed, N Jaitly, A Senior, V Vanhoucke, P Nguyen, TN Sainath, and B Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, Nov 2012. ISSN 1053-5888. doi: 10.1109/MSP.2012.2205597.

GE Hinton, S Osindero, and Y Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18, 2006.

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 Yamoto. Low-latency real-time meeting recognition and understanding using distant microphones and omni-directional camera. *IEEE Trans. Audio, Speech, Language Process.*, 20(2):499–513, 2012. doi: 10.1109/TASL.2011.2164527.

K Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251 – 257, 1991.

K Hornik, M Stinchcombe, and H White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

C-L Huang, PR Dixon, S Matsuda, Y Wu, X Lu, M Saiko, and C Hori. The NICT ASR system for IWSLT 2013. In *Proc. International Workshop on Spoken Language Translation (IWSLT)*, 2013a.

J-T Huang, J Li, D Yu, L Deng, and Y Gong. Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, 2013b.

Y Huang and Y Gong. Regularized sequence-level deep neural network model adaptation. In *Proc. ISCA Interspeech*, pages 1081–1085, 2015.

Z Huang, J Li, SM Siniscalchi, I-F Chen, J Wu, and C-H Lee. Rapid adaptation for deep neural networks through multi-task learning. In *Proc. ISCA Interspeech*, pages 3625–3629, 2015a.

Z Huang, S M Siniscalchi, I-F Chen, J Wu, and C-H Lee. Maximum a-posteriori adaptation of network parameters in deep models. *arXiv preprint arXiv:1503.02108*, 2015b.

D Hubel and T Wiesel. Receptive fields, binocular interaction, and functional architecture in the cat's visual cortex. *Journal of Physiology*, 160:106–154, 1962.

MY Hwang and X Huang. Shared-distribution hidden markov models for speech recognition. *IEEE Transactions on Speech and Audio Processing*, 1(4):414–420, Oct 1993. ISSN 1063-6676. doi: 10.1109/89.242487.

D Imseng, H Bourlard, and PN Garner. Using KL-divergence and multilingual information to improve ASR for under-resourced languages. In *Proc. IEEE Int. Conf. Acoustics, Speech Signal Processing (ICASSP)*, pages 4869–4872, 2012.

RA Jacobs, MI Jordan, SJ Nowlan, and GE Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991. doi: 10.1162/neco.1991.3.1.79. URL `http://dx.doi.org/10.1162/neco.1991.3.1.79`.

A Janin, D Baron, J Edwards, D Ellis, D Gelbart, N Morgan, B Peskin, T Pfau, E Shriberg, A Stolcke, and C Wooters. The ICSI meeting corpus. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, pages I–364–I–367, 2003.

F Jelinek. Continuous speech recognition by statistical methods. *Proceedings of the IEEE*, 64(4):532–556, April 1976. ISSN 0018-9219. doi: 10.1109/PROC.1976.10159.

VE Johnson. Revised standards for statistical evidence. *Proceedings of the National Academy of Sciences*, 110(48):19313–19317, 2013.

BH Juang and S Katagiri. Discriminative learning for minimum error classification pattern recognition. *IEEE Transactions on Signal Processing*, 40(12):3043–3054, 1992. ISSN 1053-587X. doi: 10.1109/78.175747.

J Kaiser, B Horvat, and Z Kacic. A novel loss function for the overall risk criterion based discriminative training of HMM models. In *Proc. Int. Conf. Spoken Language Processing (ICSLP)*, pages 887–890, 2000.

M Karafiat, L Burget, P Matejka, O Glembek, and J Cernozky. I-vector-based discriminative adaptation for automatic speech recognition. In *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2011.

P Karanasou, Y Wang, MJF Gales, and PC Woodland. Adaptation of deep neural network acoustic models using factorised i-vectors. In *Proc. ISCA Interspeech*, pages 2180–2184, 2014.

SM Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. In *IEEE Transactions on Acoustics, Speech and Singal processing*, volume 35, pages 400–401, March 1987.

S Kim and I Lane. Recurrent models for auditory attention in multi-microphone distance speech recognition. *CoRR*, abs/1511.06407, 2015. URL `http://arxiv.org/abs/1511.06407`.

B Kingsbury. Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, pages 3761–3764, 2009. doi: 10.1109/ICASSP.2009.4960445.

K Kinoshita, M Delcroix, S Gannot, EAP Habets, R Haeb-Umbach, W Kellermann, V Leutnant, R Maas, T Nakatani, and B Raj. A summary of the reverb challenge: state-of-the-art and remaining challenges in reverberant speech processing research. *EURASIP Journal on Advances in Signal Processing*, 2016 (1):1–19, 2016.

CH Knapp and GC Carter. The generalized correlation method for estimation of time delay. *IEEE Transactions on Acoustic, Speech and Signal Processing*, 24 (4):320–327, 1976.

R Kneser and H Ney. Improved backing-off for m-gram language modeling. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, volume I, pages 181–184, 1995.

KM Knill, MJF Gales, SP Rath, PC Woodland, C Zhang, and SX Zhang. Investigation of multilingual deep neural networks for spoken term detection. In *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 138–143, 2013.

K Kumatani, J McDonough, B Rauch, D Klakow, PN Garner, and W Li. Beamforming with a maximum negentropy criterion. *IEEE Trans. Audio, Speech, Language Process.*, 17(5):994–1008, 2009.

K Kumatani, J McDonough, and B Raj. Microphone array processing for distant speech recognition: From close-talking microphones to far-field sensors. *IEEE Signal Process. Mag.*, 29(6):127–140, 2012. ISSN 1053-5888. doi: 10.1109/MSP.2012.2205285.

P Lal. *Cross-Lingual Automatic Speech Recognition using Tandem Features*. PhD thesis, The University of Edinburgh, 2011.

P Lal and S King. Cross-lingual automatic speech recognition using tandem features. *IEEE Transactions on Audio, Speech, and Language Processing*, 21 (12):2506–2515, 2013. ISSN 1558-7916. doi: 10.1109/TASL.2013.2277932.

Y LeCun and Y Bengio. Convolutional networks for images, speech and time series. In *The Handbook of Brain Theory and Neural Networks*, pages 255–258. MIT Press, 1995.

Y LeCun, B Boser, JS Denker, D Henderson, RE Howard, W Hubbard, and LD Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1:541–551, 1989.

Y LeCun, L Bottou, Y Bengio, and P Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86:2278–2324, 1998a.

Y LeCun, L Bottou, G Orr, and K Müller. Efficient backprop. In *Neural Networks: Tricks of the Trade*, chapter 2. Springer, 1998b.

H Lee, P Pham, Y Largman, and A Ng. Unsupervised feature learning for audio classification using convolutional deep belief networks. In *Proc. Advances in Neural Information and Processing Systems (NIPS)*, pages 1096–1104, 2009.

B Li and KC Sim. Comparison of discriminative input and output transformations for speaker adaptation in the hybrid NN/HMM systems. In *Proc. ISCA Interspeech*, 2010.

G Li, H Zhu, G Cheng, K Thambiratnam, B Chitsaz, D Yu, and F Seide. Context-dependent deep neural networks for audio indexing of real-life data. In *Proc. IEEE Spoken Language Technology Workshop (SLT)*, December 2012.

J Li, J-T Huang, and Y Gong. Factorized adaptation for deep neural network. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, 2014a.

J Li, R Zhao, J-T Huang, and Y Gong. Learning small-size DNN with output-distribution-based criteria. In *Proc. ISCA Interspeech*, 2014b.

J Li, L Deng, R Haeb-Umbach, and Y Gong. *Robust Automatic Speech Recognition*. Academic Press, 2015. ISBN 978-0-12-802398-3. doi: http://dx.doi.org/10.1016/B978-0-12-802398-3.09987-6.

H Liao. Speaker adaptation of context dependent deep neural networks. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, pages 7947–7951, 2013.

H Liao, E McDermott, and A Senior. Large scale deep neural network acoustic modeling with semi-supervised training data for YouTube video transcription. In *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 368–373, 2013.

M Lincoln, I McCowan, J Vepa, and HK Maganti. The multi-channel Wall Street Journal audio visual corpus (MC-WSJ-AV): Specification and initial experiments. In *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2005.

Y Liu, P Zhang, and T Hain. Using neural network front-ends on far field multiple microphones based speech recognition. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, pages 5542–5546, 2014.

L Lu and S Renals. On training the recurrent neural network encoder-decoder for large vocabulary end-to-end speech recognition. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, 2016.

L Lu, A Ghoshal, and S Renals. Regularized subspace Gaussian mixture models for cross-lingual speech recognition. In *IEEE Signal Processing Letters*, 2011.

L Lu, A Ghoshal, and S Renals. Cross-lingual subspace gaussian mixture models for low-resource speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(1):17–27, Jan 2014. ISSN 2329-9290. doi: 10.1109/TASL.2013.2281575.

D Marino and T Hain. An analysis of automatic speech recognition with multiple microphones. In *Proc. ISCA Interspeech*, pages 1281–1284, 2011.

J Martens. Deep learning via Hessian-free optimization. In *Proc. Int. Conf. Machine Learning (ICML)*, pages 735–742, 2010.

DA Medler. A brief history of connectionism. *Neural Computing Surveys*, 1: 61–101, 1998.

P Mermelstein. Distance measures for speech recognition, psychological and instrumental. *Pattern recognition and artificial intelligence*, 116:374–388, 1976.

Y Miao and F Metze. Improving low-resource CD-DNN-HMM using dropout and multilingual DNN training. In *Proc. ISCA Interspeech*, pages 2237–2241. ISCA, 2013.

Y Miao, F Metze, and S Rawat. Deep maxout networks for low-resource speech recognition. In *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2013.

Y Miao, H Zhang, and F Metze. Speaker adaptive training of deep neural network acoustic models using i-vectors. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(11):1938–1949, Nov 2015. ISSN 2329-9290. doi: 10.1109/TASLP.2015.2457612.

T Mikolov, M Karafiát, L Burget, J Cernockỳ, and S Khudanpur. Recurrent neural network based language model. *Proc. ISCA Interspeech*, 2:3, 2010.

M Minsky and S Papert. *Perceptrons*. MIT Press, 1969.

TM Mitchell. *Machine Learning*. WCB McGraw-Hill, 1997.

A Mohamed, TN Sainath, G Dahl, B Ramabhadran, GE Hinton, and MA Picheny. Deep belief networks using discriminative features for phone recognition. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, pages 5060–5063, 2011.

A Mohamed, GE Dahl, and GE Hinton. Acoustic modeling using deep belief networks. *IEEE Transactions on Audio, Speech, and Language Processing*, 20 (1), 2012.

A Mohan and R Rose. Multi-lingual speech recognition with low-rank multi-task deep neural networks. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, pages 4994–4998, 2015. doi: 10.1109/ICASSP.2015.7178921.

N Morgan. Deep and wide: Multiple layers in automatic speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):7–13, Jan 2012. ISSN 1558-7916. doi: 10.1109/TASL.2011.2116010.

N Morgan and H Bourlard. Factoring networks by a statistical method. *Neural Computation*, 4(6):835–838, 1992.

P Motlicek, D Imseng, B Potard, P N Garner, and I Himawan. Exploiting foreign resources for DNN-based ASR. *EURASIP Journal on Audio, Speech, and Music Processing*, 2015(1):1–10, 2015.

V Nair and G Hinton. Rectified linear units improve restricted Boltzmann machines. In *Proc. Int. Conf. Machine Learning (ICML)*, pages 131–136, 2010.

J Neto, L Almeida, M Hochberg, C Martins, L Nunes, S Renals, and T Robinson. Speaker adaptation for hybrid HMM–ANN continuous speech recognition system. In *Proc. Eurospeech*, pages 2171–2174, 1995.

C Nieuwoudt and EC Botha. Cross-language use of acoustic information for automatic speech recognition. *Speech Communication*, 38(1âĂŞ2):101 – 113, 2002. ISSN 0167-6393. doi: http://dx.doi.org/10.1016/S0167-6393(01)00046-2. URL `http://www.sciencedirect.com/science/article/pii/S0167639301000462`.

T Ochiai, S Matsuda, X Lu, C Hori, and S Katagiri. Speaker adaptive training using deep neural networks. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, pages 6349–6353, 2014.

M Omologo, M Matassoni, P Svaizer, and D Giuliani. Microphone array based speech recognition with different talker-array positions. In *Proc IEEE ICASSP*, pages 227–230, 1997.

DS Pallet, WM Fisher, and JG Fiscus. Tools for the analysis of benchmark speech recognition tests. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, pages 97–100, 1990.

N Parihar, J Picone, D Pearce, and HG Hirsch. Performance analysis of the Aurora large vocabulary baseline system. In *Proc. EUSIPCO*, 2004.

S Parveen and P Green. Multitask learning in connectionist robust ASR using recurrent neural networks. In *Proc. ISCA Interspeech*, 2003.

D Povey. *Discriminative training for large vocabulary speech recognition*. PhD thesis, University of Cambridge, 2003.

D Povey, D Kanevsky, B Kingsbury, B Ramabhadran, G Saon, and K Visweswariah. Boosted MMI for model and feature-space discriminative training. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, pages 4057–4060, 2008.

D Povey, L Burget, M Agarwal, P Akyazi, F Kai, A Ghoshal, O Glembek, N Goel, M Karafiát, and A Rastrow. The subspace gaussian mixture model – a structured model for speech recognition. *Computer Speech and Language*, 25(2): 404–439, 2011a.

D Povey, A Ghoshal, G Boulianne, L Burget, O Glembek, N Goel, M Hannemann, P Motlíček, Y Qian, P Schwarz, J Silovský, G Stemmer, and K Veselý. The Kaldi speech recognition toolkit. In *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, December 2011b.

D Povey, X Zhang, and S Khudanpur. Parallel training of DNNs with natural gradient and parameter averaging. *arXiv preprint arXiv:1410.7455*, 2014.

R Price, K Iso, and K Shinoda. Speaker adaptation of deep neural networks using a hierarchy of output layers. In *Proc. IEEE Spoken Language Technology Workshop (SLT)*, 2014.

Y Qian, T Tan, and D Yu. An investigation into using parallel data for far-field speech recognition. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, 2016.

LR Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, Feb 1989. ISSN 0018-9219. doi: 10.1109/5.18626.

LR Rabiner, BH Juang, SE Levinson, and MM Sondhi. Recognition of isolated digits using hidden Markov models with continuous mixture densities. *AT&T Technical Journal*, 64(6 pt 1):1211–1234, 1985. ISSN 8756-2324.

MA Ranzato, FJ Huang, Y-L Boureau, and Y LeCun. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2007.

S Renals and P Swietojanski. Neural networks for distant speech recognition. In *Proc. Workshop on Hands-free Speech Communication and Microphone Arrays (HSCMA)*, 2014.

S Renals, N Morgan, M Cohen, and H Franco. Connectionist probability estimation in the DECIPHER speech recognition system. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, 1992.

S Renals, N Morgan, H Bourlard, M Cohen, and H Franco. Connectionist probability estimators in HMM speech recognition. *IEEE Transactions on Speech and Audio Processing*, 2:161–174, 1994.

S Renals, T Hain, and H Bourlard. Recognition and understanding of meetings: The AMI and AMIDA projects. In *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Kyoto, 12 2007. IDIAP-RR 07-46.

SJ Rennie, V Goel, and S Thomas. Annealed dropout training of deep networks. In *Proc. IEEE Spoken Language Technology Workshop (SLT)*, 2014.

M Riesenhuber and T Poggio. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2:1019–1025, 1999.

F Rosenblatt. Principles of neurodynamics: perceptrons and the theory of brain mechanisms. *Report (Cornell Aeronautical Laboratory)*, 1962.

DE Rumelhart, GE Hinton, and RJ Williams. Learning internal representations by error-propagation. In *Parallel Distributed Processing*, volume 1, pages 318–362. MIT Press, 1986.

M Saiko, H Yamamoto, R Isotani, and C Hori. Efficient multi-lingual unsupervised acoustic model training under mismatch conditions. In *Proc. IEEE Spoken Language Technology Workshop (SLT)*, pages 24–29, Dec 2014. doi: 10.1109/SLT.2014.7078544.

TN Sainath, B Kingsbury, and B Ramabhadran. Auto-encoder bottleneck features using deep belief networks. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, pages 4153–4156, 2012.

TN Sainath, B Kingsbury, A Mohamed, GE Dahl, G Saon, H Soltau, T Beran, AY Aravkin, and B Ramabhadran. Improvements to deep convolutional neural networks for LVCSR. In *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 315–320, 2013a. doi: 10.1109/ASRU.2013.6707749.

TN Sainath, B Kingsbury, H Soltau, and B Ramabhadran. Optimization techniques to improve training speed of deep neural networks for large speech tasks. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(11):2267–2276, Nov 2013b. ISSN 1558-7916. doi: 10.1109/TASL.2013.2284378.

TN Sainath, A Mohamed, B Kingsbury, and B Ramabhadran. Deep convolutional neural networks for lvcsr. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, pages 8614–8618, 2013c.

TN Sainath, RJ Weiss, KW Wilson, A Narayanan, M Bacchiani, and A Senior. Speaker location and microphone spacing invariant acoustic modeling from raw multichannel waveforms. In *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2015.

H Sakoe and S Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, Feb 1978. ISSN 0096-3518. doi: 10.1109/TASSP.1978.1163055.

L Samarakoon and K C Sim. Learning factorized feature transforms for speaker normalization. In *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 145–152. IEEE, 2015.

L Samarakoon and K C Sim. On combining i-vectors and discriminative adaptation methods for unsupervised speaker normalization in dnn acoustic models.

In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, pages 5275–5279, 2016.

G Saon, H Soltau, D Nahamoo, and M Picheny. Speaker adaptation of neural network acoustic models using i-vectors. In *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 55–59, 2013. URL `http://dblp.uni-trier.de/db/conf/asru/asru2013.html#SaonSNP13`.

T Schultz. GlobalPhone: A multilingual speech and text corpus developed at Karlsruhe University. In *Proc. ISCA Interspeech*, 2002.

T Schultz and A Waibel. Language independent and language adaptive acoustic modeling for speech recognition. *Speech Communication*, 35:31–51, 2001a.

T Schultz and A Waibel. Experiments on cross-language acoustic modeling. In *Proc. Eurospeech*, 2001b.

F Seide, X Chen, and D Yu. Feature engineering in context-dependent deep neural networks for conversational speech transcription. In *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2011.

M Seltzer and J Droppo. Multi-task learning in deep neural networks for improved phoneme recognition. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, 2013.

M Seltzer and R Stern. Subband likelihood-maximizing beamforming for speech recognition in reverberant environments. *IEEE Trans. Audio, Speech, & Lang. Process.*, 14:2109–2121, 2006.

M Seltzer, D Yu, and Y Wang. An investigation of deep neural networks for noise robust speech recognition. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, 2013.

A Senior and I Lopez-Moreno. Improving DNN speaker independence with i-vector inputs. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, pages 225–229, 2014.

A Senior, G Heigold, MA Ranzato, and K Yang. An empirical study of learning rates in deep neural networks for speech recognition. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, pages 6724–6728. IEEE, 2013.

A Senior, G Heigold, M Bacchiani, and H Liao. GMM-free DNN acoustic model training. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, pages 5602–5606, May 2014. doi: 10.1109/ICASSP.2014.6854675.

P Sermanet, S Chintala, and Y LeCun. Convolutional neural networks applied to house numbers digit classification. *CoRR*, abs/1204.3968, 2012. URL `http://arxiv.org/abs/1204.3968`.

M Sinclair, P Bell, A Birch, and F McInnes. A semi-markov model for speech segmentation with an utterance-break prior. In *Proc. ISCA Interspeech*, 2014.

SM Siniscalchi, J Li, and CH Lee. Hermitian polynomial for speaker adaptation of connectionist speech recognition systems. *IEEE Trans Audio, Speech, and Language Processing*, 21:2152–2161, 2013. ISSN 1558-7916. doi: 10.1109/TASL.2013.2270370.

O Siohan. Sequence training of multi-task acoustic models using meta-state labels. In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, pages 5425–5429, 2016. doi: 10.1109/ICASSP.2016.7472714.

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. URL `http://jmlr.org/papers/v15/srivastava14a.html`.

A Stan, P Bell, and S King. A grapheme-based method for automatic alignment of speech and text data. In *Proc. IEEE Spoken Language Technology Workshop (SLT)*, 2012.

SS Stevens, J Volkmann, and EB Newman. A Scale for the Measurement of the Psychological Magnitude Pitch. 8(3):185–190, January 1937. doi: 10.1121/1.1915893. URL `http://dx.doi.org/10.1121/1.1915893`.

A Stolcke. Making the most from multiple microphones in meeting recognition. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, 2011.

A Stolcke, F Grézl, M-Y Hwang, X Lei, N Morgan, and D Vergyri. Cross-domain and cross-language portability of acoustic features estimated by multilayer perceptrons. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, 2006. doi: 10.1109/ICASSP.2006.1660022.

A Stolcke, X Anguera, K Boakye, O Cetin, A Janin, M Magimai-Doss, C Wooters, and J Zheng. The SRI-ICSI Spring 2007 meeting and lecture recognition system. In R Stiefelhagen, R Bowers, and J Fiscus, editors, *Multimodal Technologies for Perception of Humans*, number 4625 in LNCS, pages 373–389. Springer, 2008.

I Sutskever, J Martens, G Dahl, and G Hinton. On the importance of initialization and momentum in deep learning. In *Proc. Int. Conf. Machine Learning (ICML)*, volume 28, pages 1139–1147, 2013. URL `http://jmlr.org/proceedings/papers/v28/sutskever13.pdf`.

P Swietojanski and S Renals. Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models. In *Proc. IEEE Spoken Language Technology Workshop (SLT)*, 2014.

P Swietojanski and S Renals. Differentiable pooling for unsupervised speaker adaptation. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, pages 4305–4309, 2015.

P Swietojanski and S Renals. Differentiable Pooling for Unsupervised Acoustic Model Adaptation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(10):1773–1784, Oct 2016. ISSN 2329-9290. doi: 10.1109/TASLP.2016.2584700.

P Swietojanski and S Renals. SAT-LHUC: Speaker adaptive training for learning hidden unit contributions. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, pages 5010–5014, 2016.

P Swietojanski, A Ghoshal, and S Renals. Unsupervised cross-lingual knowledge transfer in DNN-based LVCSR. In *Proc. IEEE Spoken Language Technology Workshop (SLT)*, pages 246–251, Miami, Florida, USA, December 2012. doi: 10.1109/SLT.2012.6424230.

P Swietojanski, A Ghoshal, and S Renals. Revisiting hybrid and GMM-HMM system combination techniques. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, 2013a.

P Swietojanski, A Ghoshal, and S Renals. Hybrid acoustic models for distant and multichannel large vocabulary speech recognition. In *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2013b. doi: 10.1109/ASRU.2013.6707744.

P Swietojanski, A Ghoshal, and S Renals. Convolutional neural networks for distant speech recognition. *IEEE Signal Processing Letters*, 21(9):1120–1124, September 2014a. ISSN 1070-9908. doi: 10.1109/LSP.2014.2325781.

P Swietojanski, J Li, and J-T Huang. Investigation of maxout networks for speech recognition. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, 2014b.

P Swietojanski, P Bell, and S Renals. Structured output layer with auxiliary targets for context-dependent acoustic modelling. In *Proc. ISCA Interspeech*, pages 3605–3609, 2015.

P Swietojanski, J Li, and S Renals. Learning hidden unit contributions for unsupervised acoustic model adaptation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(8):1450–1463, Aug 2016. ISSN 2329-9290. doi: 10.1109/TASLP.2016.2560534.

T Tan, Y Qian, M Yin, Y Zhuang, and K Yu. Cluster adaptive training for deep neural network. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, 2015.

TheScipyCommunity. *NumPy Reference Guide*. SciPy.org, 2016. URL `http://docs.scipy.org/doc/numpy/reference/`.

S Thomas, S Ganapathy, and H Hermansky. Cross-lingual and multi-stream posterior features for low resource LVCSR systems. In *Proc. ISCA Interspeech*, 2010.

S Thomas, S Ganapathy, and H Hermansky. Multilingual MLP features for low-resource LVCSR systems. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, 2012a.

S Thomas, S Ganapathy, A Jansen, and H Hermansky. Data-driven posterior features for low resource speech recognition applications. In *Proc. ISCA Interspeech*, 2012b.

N Tomashenko and Y Khokhlov. GMM-derived features for effective unsupervised adaptation of deep neural network acoustic models. In *Proc. ISCA Interspeech*, 2015.

L Toth. Convolutional deep maxout networks for phone recognition. In *Proc. ISCA Interspeech*, 2014.

E Trentin. Networks with trainable amplitude of activation functions. *Neural Networks*, 14:471–493, 2001.

J Trmal, J Zelinka, and L Müller. On speaker adaptive training of artificial neural networks. In *Proc. ISCA Interspeech*, 2010.

AM Turing. On computable numbers, with an application to the entscheidungsproblem. *Journal of Mathematics*, 58(345-363):5, 1936.

AM Turing. Computing machinery and intelligence. *Mind Association*, 59(236): 433–460, 1950.

D Van Compernolle, W Ma, F Xie, and M Van Diest. Speech recognition in noisy environments with the aid of microphone arrays. *Speech Commun.*, 9:433–442, 1990.

E van den Berg, B Ramabhadran, and M Picheny. Neural network training variations in speech and subsequent performance evaluation. In *Int. Conf. Learning Representations (ICLR), Workshop track*, 2016.

LJP van der Maaten and GE Hinton. Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research*, 9:2579âĂŞ2605, 2008.

VM Velichko and NG Zagoruyko. Automatic recognition of 200 words. *International Journal of Man-Machine Studies*, 2(3):223 – 234, 1970.

K Vesely, A Ghoshal, L Burget, and D Povey. Sequence-discriminative training of deep neural networks. In *Proc. ISCA Interspeech*, pages 2345–2349, 2013a.

K Vesely, M Hannemann, and L Burget. Semi-supervised training of deep neural networks. In *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 267–272, 2013b.

TK Vintsyuk. Speech discrimination by dynamic programming. *Cybernetics*, 4 (1):52–57, 1968. Russian Kibernetika 4(1):81-88 (1968).

A Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, April 1967. ISSN 0018-9448. doi: 10.1109/TIT.1967.1054010.

NT Vu, F Kraus, and T Schultz. Cross-language bootstrapping based on completely unsupervised training using multilingual A-stabil. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, 2011. doi: 10.1109/ICASSP.2011.5947479.

A Waibel, T Hanazawa, G Hinton, K Shikano, and KJ Lang. Phoneme recognition using time-delay neural networks. *IEEE Trans Acoust., Speech, and Signal Process.*, 37:328–339, 1989.

G Wang and KC Sim. Regression-based context-dependent modeling of deep neural networks for speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(11):1660–1669, Nov 2014. ISSN 2329-9290. doi: 10.1109/TASLP.2014.2344855.

M Wester, Z Wu, and J Yamagishi. Human vs machine spoofing detection on wideband and narrowband data. In *Proc. ISCA Interspeech*, pages 2047–2051, 2015.

S Wisdom, JR Hershey, J Le Roux, and S Watanabe. Deep unfolding for multichannel source separation. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, 2016.

M Wölfel and J McDonough. *Distant Speech Recognition*. Wiley, 2009.

M Wölfel, C Fügen, S Ikbal, and J McDonough. Multi-source far-distance microphone selection and combination for automatic transcription of lectures. In *Proc. Int. Conf. Spoken Language Processing (ICSLP)*, 2006.

PC Woodland. Speaker adaptation for continuous density HMMs: A review. In *Proceedings of the ISCA workshop on adaptation methods for speech recognition*, pages 11–19, 2001.

C Wu and M Gales. Multi-basis adaptive neural network for rapid adaptation in speech recognition. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, 2015.

X Xiao, S Watanabe, H Erdogan, L Lu, JR Hershey, ML Seltzer, G Chen, Y Zhang, M Mandel, and D Yu. Deep beamforming networks for multi-channel speech recognition. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, 2016.

J Xue, J Li, D Yu, M Seltzer, and Y Gong. Singular value decomposition based low-footprint speaker adaptation and personalization for deep neural network. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, pages 6359–6363, 2014a.

S Xue, O Abdel-Hamid, J Hui, L Dai, and Q Liu. Fast adaptation of deep neural network based on discriminant codes for speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(12):1713–1725, Dec 2014b. ISSN 2329-9290. doi: 10.1109/TASLP.2014.2346313.

K Yao, D Yu, F Seide, H Su, L Deng, and Y Gong. Adaptation of context-dependent deep neural networks for automatic speech recognition. In *Proc. IEEE Spoken Language Technology Workshop (SLT)*, pages 366–369, 2012.

T Yoshioka, A Sehr, M Delcroix, K Kinoshita, R Maas, T Nakatani, and W Kellermann. Making machines understand us in reverberant rooms: Robustness against reverberation for automatic speech recognition. *IEEE Signal Process. Mag.*, 29(6):114–126, 2012.

T Yoshioka, N Ito, M Delcroix, A Ogawa, K Kinoshita, M Fujimoto, C Yu, WJ 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. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 436–443, Dec 2015. doi: 10.1109/ASRU.2015.7404828.

S Young. Talking to machines. *Royal Academia of Engineering - Ingenia*, (9), 2013.

SJ Young and PC Woodland. State clustering in hidden Markov model-based continuous speech recognition. *Computer Speech and Language*, 8(4):369–383, 1994.

D Yu and L Deng. *Automatic Speech Recognition: A Deep Learning Approach*. Springer Publishing Company, Incorporated, 2014. ISBN 1447157788, 9781447157786.

D Yu, M Seltzer, J Li, J-T Huang, and F Seide. Feature learning in deep neural networks - studies on speech recognition. In *Proc. Int. Conf. Learning Representations (ICLR)*, 2013a. URL `http://research.microsoft.com/apps/pubs/default.aspx?id=189337`.

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. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, pages 7893–7897, 2013b.

MD Zeiler and R Fergus. Differentiable pooling for hierarchical feature learning. *CoRR*, abs/1207.0151, 2012. URL `http://arxiv.org/abs/1207.0151`.

A Zgank, Z Kacic, and B Horvat. Comparison of acoustic adaptation methods in multilingual speech recognition environment. In *Proc. ICTSD*, volume 6, pages 245–250, 2003.

C Zhang and PC Woodland. Standalone training of context-dependent deep neural network acoustic models. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, 2014a.

C Zhang and PC Woodland. Context independent discriminative pre-training. Unpublished work, 2014b.

C Zhang and PC Woodland. Parameterised sigmoid and ReLU hidden activation functions for DNN acoustic modelling. In *Proc. ISCA Interspeech*, pages 3224–3228, 2015.

X Zhang, J Trmal, D Povey, and S Khudanpur. Improving deep neural network acoustic models using generalized maxout networks. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, 2014.

Y Zhang, G Chen, D Yu, K Yao, S Khudanpur, and J Glass. Highway long short-term memory rnns for distant speech recognition. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, 2016.

Y Zhao, J Li, J Xue, and Y Gong. Investigating online low-footprint speaker adaptation using generalized linear regression and click-through data. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, pages 4310–4314, 2015.

G Zweig and P Nguyen. A segmental CRF approach to large vocabulary continuous speech recognition. In *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 152–157, 2009. doi: 10.1109/ASRU.2009.5372916.

E Zwyssig, F Faubel, S Renals, and M Lincoln. Recognition of overlapping speech using digital MEMS microphone arrays. In *Proc. IEEE Int. Conf. Acoustic, Speech Signal Processing (ICASSP)*, 2013.