View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Infoscience - École polytechnique fédérale de Lausanne

Towards End-to-End Speech Recognition

THÈSE N^{O} 7054 (2016)

PRÉSENTÉE LE 13 JUILLET 2016
À LA FACULTÉ DES SCIENCES ET TECHNIQUES DE L'INGÉNIEUR
LABORATOIRE DE L'IDIAP
PROGRAMME DOCTORAL EN GÉNIE ÉLECTRIQUE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Dimitri PALAZ

acceptée sur proposition du jury:

Dr J.-M. Vesin, président du jury
Prof. H. Bourlard, Dr R. Collobert, directeurs de thèse
Prof. S. Renals, rapporteur
Dr S. Bengio, rapporteur
Prof. J.-Ph. Thiran, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE
Suisse
2016

To my parents.

Acknowledgements

First, I would like to thank my co-supervisors, Ronan and Mathew, for their invaluable support, ideas, advices, help and teachings they provided me during these four years. I specially thank Ronan for sharing his views and for offering me the chance to work on deep learning. I specially thank Mathew for his dedication and his invaluable help. I also thank Hervé for giving me the opportunity to work at Idiap on this project. Thanks also to the Hasler Foundation who supported me financially.

Secondly, I am grateful to my thesis committee, Samy Bengio, Steve Renals, Jean-Philippe Thiran and Jean-Marc Vesin for reviewing this thesis and providing constructive comments and suggestions.

A special thanks to my colleagues of the AML group, Joel, Remi and Pedro, my brothers in arms, for our discussions, for their support, help and advices and for the amazing time we spend together. I also would like to thank my other colleagues at Idiap, specially Bastien and Laurent for our always enjoyable breaks and Nikos for our insightful discussions. Thanks also to Marc, Christian, Cesar, Ramya, Marzieh, David, Phil, Pierre-Edouard, Raphael, Srikanth, Gulcan, Guillaume, Frank, Afsaneh, Milos, Norbert, Ivan, Salim, Louis-Marie, Thomas, Olivia, Andrej, Ilya, Arjan, Leo, Darshan, Nesli, Hannah, Afroze, Elie, Gwénolé, Nadine, Sylvie and others.

Special thanks to Dr. Daniel Egger, for his moral support, as well as Dominique, Leo, Nicolas, Vincent and Zoé also for their moral support. Finally, thanks to my parents to whom I owe everything.

Lausanne, 11 June 2016 Dimitri Palaz

Abstract

Standard automatic speech recognition (ASR) systems follow a divide and conquer approach to convert speech into text. Alternately, the end goal is achieved by a combination of sub-tasks, namely, feature extraction, acoustic modeling and sequence decoding, which are optimized in an independent manner. More recently, in the machine learning community deep learning approaches have emerged which allow training of systems in an *end-to-end* manner. Such approaches have found success in the area of natural language processing and computer vision community, and have consequently peaked interest in the speech community. The present thesis builds on these recent advances to investigate approaches to develop speech recognition systems in end-to-end manner. In that respect, the thesis follows two main axes of research. The first axis of research focuses on joint learning of features and classifiers for acoustic modeling. The second axis of research focuses on joint training of the acoustic model and the decoder, leading to an end-to-end sequence recognition system.

Along the first axis of research, in the framework of hybrid hidden Markov model/artificial neural networks (HMM/ANN) based ASR, we develop a convolution neural networks (CNNs) based acoustic modeling approach that takes raw speech signal as input and estimates phone class conditional probabilities. Specifically, the CNN has several convolution layers (feature stage) followed by multilayer perceptron (classifier stage), which are jointly optimized during the training. Through ASR studies on multiple languages and extensive analysis of the approach, we show that the proposed approach, with minimal prior knowledge, is able to learn automatically the relevant features from the raw speech signal. This approach yields systems that have less number of parameters and achieves better performance, when compared to the conventional approach of cepstral feature extraction followed by classifier training. As the features are automatically learned from the signal, a natural question that arises is: are such systems robust to noise? Towards that we propose a robust CNN approach referred to as normalized CNN approach, which yields systems that are as robust as or better than the conventional ASR systems using cepstral features (with feature level normalizations).

The second axis of research focuses on end-to-end sequence recognition. We first propose an end-to-end phoneme recognition system. In this system the relevant features, classifier and the decoder (based on conditional random fields) are jointly modeled during training. We demonstrate the viability of the approach on TIMIT phoneme recognition task. Building on top of that, we investigate a “weakly supervised” training that alleviates the necessity for frame level alignments. Finally, we extend the weakly supervised approach to propose a novel keyword spotting technique. In this technique, a CNN first process the input observation sequence to output word level scores, which are subsequently aggregated to detect or spot words. We demonstrate the potential of the approach through a comparative study on LibriSpeech with the standard approach of keyword word spotting based on lattice indexing using ASR system.

Key words: Deep learning, automatic speech recognition, end-to-end training, convolutional neural networks, raw speech signal, robust speech recognition, conditional random fields, weakly-supervised training, keyword spotting.

Résumé

Les systèmes de reconnaissance automatique de la parole (RAP) standard suivent une approche basée sur l'adage "Diviser pour mieux régner" pour convertir de la parole en texte. Autrement dit, le but final est atteint par une combinaison de sous-tâches, plus précisément : l'extraction de représentations, la modélisation acoustique et le décodage de séquence. Ces sous-tâches sont optimisées indépendamment. Récemment, dans la communauté de l'apprentissage automatique, des approches d'apprentissage profonds ont été développées, permettant d'entrainer des systèmes "de bout en bout". Ces approches ont été fructueuses dans les domaines du traitement automatique des langues et de la vision par ordinateur et ont par conséquent attirés l'attention de la communauté en reconnaissance de la parole. Cette thèse se base sur ces avancées récentes pour étudier l'application de l'entrainement "de bout en bout" aux systèmes de reconnaissance de la parole. A cet égard, cette thèse suit deux axes de recherche. Le premier axe se focalise sur l'apprentissage joint des représentations et de la modélisation acoustique. Le deuxième axe se focalise sur la modélisation jointe du modèle acoustique et du décodage de séquence.

Suivant le premier axe, dans le cadre de la RAP basée sur l'approche hybride HMM/ANN, nous développons une approche de modèle acoustique basée sur les réseaux de neurones à convolution, qui prennent en entrée le signal audio brut et estiment les probabilités conditionnelles des classes phonétiques. Plus précisément, le réseau est composé de plusieurs couches d'apprentissage de représentation, suivi de couches de modélisation acoustique, implémentées par un perceptron multicouche. Toutes les couches sont entrainées conjointement. Au travers de plusieurs études de RAP sur différentes langues et d'analyses étendues, nous montrons que l'approche proposée est capable d'apprendre automatiquement des représentations pertinentes à partir du signal brut, en utilisant un minimum de connaissance préalable. Les systemes basés sur cette approche sont tout aussi performant que les systèmes classiques basés sur l'extraction de représentations cepstrales, en ayant moins de paramètres. Étant donné que les représentations sont apprises automatiquement, on peut se poser la question de la robustesse au bruit de ces systèmes. Dans cette direction, nous proposons une approche robuste basée sur les réseaux à convolution, nommée réseaux à convolution normalisés. Nous montrons que les systèmes basés sur cette approche sont aussi robustes que les systèmes conventionnels basés sur le renforcement des représentations cepstrales.

Le deuxième axe de recherche se focalise sur la conversion de séquence à séquence, entrainés de bout en bout. Premièrement, nous proposons un système de reconnaissance de séquence de phonème, entrainé de bout en bout. Dans ce système, les représentations, la classification et le décodage de séquence, basé sur des *Conditional Random Fields*, sont modélisées conjointement pendant l'entrainement. Nous démontrons la viabilité de cette approche sur une tache de reconnaissance de phonème. Sur ces bases, nous étudions un entrainement faiblement supervisé, qui permet d'éliminer l'utilisation d'alignement temporel. Finalement, nous proposons une technique novatrice de détection de mot-clés basée sur l'approche d'entrainement faiblement supervisé. Dans cette technique, un réseau à convolution traite la séquence d'entrée pour obtenir des scores au niveau des mots. Ces scores ont ensuite agrégés pour détecter ou repérer des mots. Nous démontrons le potentiel de cette approche au travers d'une étude comparative sur LibriSpeech avec un système de référence standard, basé sur l'indexation des treillis.

Mots clefs : Apprentissage automatique profond, reconnaissance automatique de la parole, entrainement de bout en bout, réseaux de neurones à convolution, signal brut de parole, reconnaissance robuste de la parole, entrainement faiblement supervisé, détection de mot-clés.

Contents

Acknowledgements **v**

Abstract (English/Français) **vii**

List of figures **xv**

List of tables **xvii**

List of acronyms **xix**

Notations **xxi**

1 Introduction **1**
- 1.1 Motivations and Objectives 1
- 1.2 Contributions 2
- 1.3 Organization of the Thesis 4

2 Background **7**
- 2.1 Overview 7
- 2.2 Features 8
- 2.3 HMM-based Speech Recognition 9
  - 2.3.1 Acoustic Likelihood Estimation 10
  - 2.3.2 Lexicon 11
  - 2.3.3 Language Model 12
  - 2.3.4 Decoding and Evaluation 13
- 2.4 Neural Networks-based Acoustic Modeling 13
  - 2.4.1 Artificial Neural Networks 13
  - 2.4.2 Architectures 15
- 2.5 Sequence-to-sequence Conversion 17
  - 2.5.1 Long Short Term Memory 18
  - 2.5.2 Conditional Random Fields 18
- 2.6 Keyword Spotting 19
  - 2.6.1 Approaches 19
  - 2.6.2 Metric 20
- 2.7 Summary 21

3 CNN-based ASR using Raw Speech Signal as Input 23
- 3.1 Related Work 24
- 3.2 Proposed CNN-based Approach 25
  - 3.2.1 Convolutional Neural Networks 26
  - 3.2.2 Network Training 28
  - 3.2.3 Illustration of a Trained Network 28
- 3.3 Recognition Studies 29
  - 3.3.1 Databases and Setup 30
  - 3.3.2 Systems 31
  - 3.3.3 Results 33
- 3.4 Summary 36

4 Analysis of Proposed CNN-based System 37
- 4.1 First Convolution Layer 37
  - 4.1.1 Input level Analysis 37
  - 4.1.2 Learned Filters 38
  - 4.1.3 Response of Filters to Input Speech Signal 42
- 4.2 Intermediate Feature level Analysis 47
  - 4.2.1 Discriminative Features 47
  - 4.2.2 Cross-domain and Cross-lingual Studies 47
- 4.3 Relation to Recent Literature 50
- 4.4 Summary 51

5 Deep Features and Shallow Classifier 53
- 5.1 Architecture and Network Design 54
- 5.2 Experimental Setup 55
- 5.3 Results 56
  - 5.3.1 Phoneme Recognition Study 56
  - 5.3.2 Continuous Speech Recognition Study 57
- 5.4 Discussion and Summary 57

6 Towards Noise-Robust Raw Speech-based Systems 59
- 6.1 Related Literature 59
- 6.2 Normalized Convolutional Neural Networks (NCNN) 60
  - 6.2.1 Normalization Layer 61
  - 6.2.2 Rectifier Linear Unit 61
- 6.3 Connected Word Recognition Study 62
  - 6.3.1 Database 62
  - 6.3.2 Baselines 63
  - 6.3.3 CNN-based Systems 63
  - 6.3.4 Results 63
- 6.4 Continuous Speech Recognition 66
  - 6.4.1 Database 66

6.4.2 Baselines 66
6.4.3 CNN-based Systems 66
6.4.4 Results 67
6.5 Analysis 67
6.5.1 Architectures Analysis 68
6.5.2 First Convolution Layer Analysis 69
6.5.3 Waveform Enhancement Study 70
6.6 Summary 72

7 End-to-end Phoneme Sequence Recognition 73
7.1 End-to-end Sequence Recognition 74
7.1.1 Supervised Training 75
7.1.2 Weakly-supervised Training 76
7.2 Phoneme Sequence Recognition Study 78
7.2.1 Experimental Setup 79
7.2.2 Results 80
7.3 Discussion 82
7.3.1 Analysis and Assessment of the Proposed Approach 82
7.3.2 Relation to Global Training Methods 82
7.3.3 Relation to Connectionist Temporal Classification 83
7.3.4 Relation to Sequence-discriminative Approaches for Acoustic Modeling 84
7.4 Summary 85

8 Jointly Learning to Locate and Classify Words 87
8.1 Related Work 87
8.2 Proposed Approach 88
8.2.1 Two-stage CNN-based System 88
8.2.2 Training 89
8.2.3 Inference 90
8.3 Experimental Setup 91
8.3.1 Database 91
8.3.2 Proposed System Setup 91
8.3.3 Evaluation Studies 92
8.4 Results 94
8.4.1 Word localisation study 94
8.4.2 Keywords Spotting Study 94
8.5 Analysis 95
8.6 Summary 96

9 Conclusions 97
9.1 Direction of Future Research 99

Bibliography 101

Curriculum Vitae **119**

List of Figures

1.1 Illustration of the incremental approach: (a) standard system, (b) joint feature and classifier training and (c) end-to-end phoneme sequence recognition. . . . 3

2.1 Overview of a general ASR system. 7
2.2 MFCC and PLP extraction pipelines. |DFT| denotes the magnitude of the discrete Fourier transform, DCT denotes the magnitude of the discrete cosine transform, AR modeling stands for auto-regressive modeling, Δ and $\Delta\Delta$ denote the first and second order derivatives across time, respectively. 9
2.3 Bloc diagram of the Viterbi EM approach. 12

3.1 Typical CNN-based system using Mel filterbanks coefficient as input [Swietojanski et al., 2014]. 23
3.2 Overview of the proposed CNN-based approach. 25
3.3 Overview of the convolutional neural network architecture. Several stages of convolution/pooling/tanh might be considered. Our network included 3 stages. The classification stage can have multiple hidden layers. 26
3.4 Illustration of a convolutional layer. d_{in} and d_{out} are the dimension of the input and output frames. kW is the kernel width (here $kW = 3$) and dW is the shift between two linear applications (here, $dW = 2$). 27
3.5 Illustration of max-pooling layer. kW_{mp} is the number of frame taken for each max operation (here, $kW_{mp} = 2$) and d represents the dimension of input/output frames (which are equal). In this case, the shift $dW_{mp} = kW_{mp}$. 27
3.6 Illustration of the feature stage of CNN trained on TIMIT to classify 183 phoneme classes. κ and δ indicates the temporal information modeled by the layer and the shift respectively. Non-linearity layers are applied after each max-pooling. . 29

4.1 Examples of three close pairs of filters learned. The left column is from CNN-1H WSJ, the center one is from CNN-1H MP-DE, the right one is from CNN-1H MP-FR. 39
4.2 Cumulative frequency responses of the learned filterbank on WSJ, MP-DE and MP-FR. 41
4.3 Normalized energy output of each filter in the first convolution layer of CNN-1H-mono WSJ for an input speech segment corresponding to phoneme $/I/$. . . 43
4.4 Mean frequency response on the WSJ-mono corpus for phonemes $/E/$, $/A/$, $/O/$, $/I/$ and $/U/$. 44

4.5 Mean frequency response on the MP-DE corpus for phonemes $/E/$, $/A/$, $/O/$, $/I/$ and $/U/$. 44
4.6 Mean frequency response on the MP-FR corpus for phonemes $/E/$, $/A/$, $/O/$, $/I/$ and $/U/$. 45
4.7 Mean frequency response for English, German and French for phoneme $/I/$. . . 45
4.8 Mean frequency response for English, German and French for phoneme $/A/$. . . 46
4.9 Illustration of the cross-domain experiment. The filter stage is trained on domain 1, then used as feature extractor on domain 2. 48

5.1 Convolutional neural network based architecture using a linear classifier. . . . 54

6.1 Illustration of the normalized convolutional neural network architecture with N filter stages. 61
6.2 Illustration of the first filter stage of the normalized convolutional neural network. 62
6.3 Cumulative frequency responses on the Aurora2 corpus on (a) clean training, (b) multi-conditional training. 70
6.4 Cumulative frequency responses on the Aurora4 corpus on (a) clean training and (b) multi-conditional training. 71
6.5 Comparison of recognition performance on the test set A of Aurora2 using original and enhanced waveforms, on (a) the clean training setup and (b) the multi-conditional training setup. 71

7.1 Illustration of proposed system. 74
7.2 Illustration of the CRF graph for 3 classes. 75
7.3 Illustration of the two training strategies: (a) separate training and (b) joint training. 76
7.4 Illustration of the cyclic graph for 3 classes. 77
7.5 Illustration of the acyclic expanded graph for 3 classes, with $t_{min} = 3$ and $t_{max} = 5$. 77
7.6 Illustration of the ANN-based system using MFCC features as input. 80
7.7 Phoneme segmentation example using the 39 phoneme set, for sequence `sx32` of speaker `mcdc0`. 81

8.1 Illustration of the proposed system. The gray input frames represent the padding. 88
8.2 Mean IoU for each word on the *test_clean* set. 94
8.3 Illustration of an inferred sequence on the top and its corresponding ground-truth, on the bottom. 95

List of Tables

3.1 Range of hyper parameters considered for the grid search. 33

3.2 Architecture of CNN-based system for different tasks. HL=1 denotes CNN-1H and HL=3 denotes CNN-3H. w_{in} is expressed in terms of milliseconds. The hyper-parameters kW, dW, d_{out} and kW_{mp} for each convolution layer is comma separated. HU denotes the number of hidden units. 3 × 1000 means 1000 hidden units per hidden layer. 33

3.3 Phoneme error rate of different systems on the core test set of the TIMIT corpus. 34

3.4 Phoneme error rate of different systems reported in literature on the core test set of the TIMIT corpus. 35

3.5 Word Error Rate on the Nov'92 testset of the WSJ corpus 35

3.6 Word Error Rate on the testset of the MP-DE corpus. 35

3.7 Word Error Rate on the testset of the MP-FR corpus. 36

4.1 Single layer perceptron based system results on the Nov'92 testset of the WSJ task. 47

4.2 Cross-domain results on English. The TIMIT results are in terms of PER. The WSJ task results are in terms of WER. 49

4.3 Crosslingual studies result on English, German and French. The feature stage is learned on Domain 1 and the classifier stage is learned on Domain 2. 50

5.1 Network hyper-parameters. 56

5.2 Results on the TIMIT core testset with 64k parameters. 57

5.3 Results on the TIMIT core testset with 132k and 320k parameters. 57

5.4 Results on the Nov'92 testset of the WSJ corpus. 58

6.1 Architecture of CNN-based system for the Aurora2 studies. w_{in} is expressed in terms of milliseconds. The hyper-parameters kW, dW, d_{out} and kW_{mp} for each convolution layer is comma separated. HU denotes the number of hidden units. 63

6.2 Comparison with the literature on Aurora2 multi-conditional training setup, expressed in Word recognition rate (WRR). DAE stands for denoising auto-encoder and DVAT stands for discriminative adaptive training using vector Taylor series. 64

6.3 Word recognition rate (WRR) on the Aurora2 test sets. HMM/GMM baseline performance using MFCC are reported in [Hirsch and Pearce, 2000] and the HMM/GMM baseline performance using MFCC-CMVN are reported in [Garner, 2009]. 65

6.4 Word recognition rate (WRR) on the Aurora2 test sets, using enhanced waveforms. HMM/GMM baseline performance using MFCC are reported in [Hirsch and Pearce, 2006]. 65
6.5 Architecture of CNN-based system for the Aurora4 studies. 67
6.6 Word recognition rate of the Aurora4 test sets on the clean training setup. . . . 68
6.7 Word recognition rate of the Aurora4 test sets on the multi-conditional training setup. 68
6.8 Comparison with literature on the multi-conditional training setup. 68
6.9 Word accuracy on the Aurora2 corpus for different normalization strategies. . . 69
6.10 Word accuracy on the Aurora2 corpus for different non-linearities. 69

7.1 Network hyper-parameters. 79
7.2 Evaluation of the proposed approach on the TIMIT core testset. Results are expressed in terms of PER. The CRF baseline performance is reported in [Morris and Fosler-Lussier, 2008] and the ML-CRF performance is reported in [Prabhavalkar and Fosler-Lussier, 2010]. 81
7.3 Comparison of global training methods. RAW denotes the use of raw speech as input, SA denotes the segment-based classification, JALM denotes the joint training of the acoustic and language model and SL denotes the segmentation learning. 83

8.1 Keywords list (in vocabulary). 93
8.2 Word position accuracies. 94
8.3 Keyword spotting performance on the *test_clean* and the *test_clean* set of LibriSpeech. 95
8.4 Nearest neighbors examples (in column). 96

List of Acronyms

AFE	Advanced front-end
ANN	Artificial neural network
ASR	Automatic speech recognition
CMVN	Cepstral mean and variance normalization
CNN	Convolutional neural network
CRF	Conditional random fields
DNN	Deep neural networks
HMM	Hidden Markov model
HMM/ANN	Hidden Markov model system using artificial neural network as acoustic model
HMM/GMM	Hidden Markov model system using Gaussian mixture models as acoustic model
HTK	HMM toolkit
KWS	Keyword spotting
LSTM	Long Short Term Memory
MFCC	Mel frequency cepstral coefficients
MLP	Multilayer perceptron
MP-DE	Mediaparl Swiss German corpus
MP-FR	Mediaparl Swiss French corpus
MTWV	Maximum term weight value
NCNN	Normalized convolutional neural network
NIST	National Institute of Standards and Technology
PER	Phoneme error rate
PLP	Perceptual linear prediction coefficients
RNN	Recurrent neural network
ROC	Receiver operating curve
SLP	Single layer perceptron
SNR	Signal-to-noise ratio
WER	Word error rate
WRR	Word recognition rate
WSJ	Wall street journal

Notations

$S = \{\mathbf{s}_1^c \dots \mathbf{s}_t^c \dots \mathbf{s}_T^c\}$	Raw speech utterance
$\mathbf{s}_t^c = \{s_{t-c} \dots s_t \dots s_{t+c}\}$	Raw speech segment at frame t of length $2c$
$X = \{\mathbf{x}_1 \dots \mathbf{x}_t \dots \mathbf{x}_T\}$	Features sequence of length T
$\mathbf{x}_t$	Features frame at time t
W	Word sequence
Θ	Parameters set
kW	Kernel width of the convolution layer
dW	Shift of the convolution layer
kW_{mp}	Kernel width of the max-pooling layer
dW_{mp}	Shift of the max-pooling layer
$\mathcal{L}$	Likelihood

Notation specific to Chapter 7

θ_f	Acoustic model parameters set
θ_A	CRF parameters set
$f_t^i(S,\Theta)$	Output of the acoustic model at time t for phoneme class i
$L = \{l_1 \dots l_t \dots l_T\}$	Phoneme label segmentation
$\Lambda = \{\lambda_1 \dots \lambda_n \dots \lambda_N\}$	Phoneme transcription
A	Phoneme transition matrix
$c(S, L, \Theta)$	CRF score path for input sequence S, label path L and parameters Θ
$\mathcal{U}_T$	Fully-connected CRF graph of length T
$\mathcal{C}_T$	Constrained CRF graph of length T
t_{max}	Maximum time a path can stay in the same label
t_{min}	Minimum time a path can stay in the same label

Notation specific to Chapter 8

$\mathcal{D}$	Dictionary
$\phi_t^w(X)$	Localisation score at time t for word w
$\Phi^w(X)$	Detection score for input sequence X
y_w	BoW label for word w

1 Introduction

This thesis takes place in the context of Automatic Speech Recognition (ASR). The goal of automatic speech recognition systems is to convert a speech signal into text. Standard ASR systems divide this task into several sub-tasks, which are optimized in an independent manner. In a first step, the speech signal is transformed into features, based on speech production and auditory knowledge. In a second step, the relationship between the features and linguistic units, such as phoneme, is modeled by estimating the acoustic likelihood. Finally, searching for the most probable word hypothesis from the acoustic likelihood estimation under syntactical and lexical constraints. This “divide and conquer” strategy has great advantages: features extraction lead to “good” representation for the task, using linguistic units allows a flexible lexicon and helps estimating the acoustic likelihood. Finally, such decomposition of a problem considerably reduces the computational cost, each step being processed separately. However, this approach could lead to sub-optimal systems. In other fields of research, e.g. text processing, computer vision, it has been shown that learning sub-tasks *jointly* can yield better systems when compared to the “divide and conquer” approach. In this thesis, we question the “divide and conquer” approach of the standard ASR systems.

1.1 Motivations and Objectives

Recent advances in machine learning have made possible systems that can be trained in an end-to-end manner, i.e. systems where every step is *learned* simultaneously, taking into account all the other steps and the final task of the whole system. It is usually referred to as *deep learning*, mainly because such architectures are usually composed of many layers (supposed to provide an increasing level of abstraction), compared to classical “shallow” systems. In contrast to “divide and conquer” approach (where each step is independently optimized) this approach has the potential to lead to more optimal systems. In the literature, recent work by Collobert et al. [2011b] presents a good illustration of this idea applied to Natural Language Processing (NLP). In that study, the authors proposed a deep neural network, which learns the word representation (the features) and the alignment discriminatively in an end-to-end manner for various NLP tasks, such as part of speech tagging, name entity recognition or semantic role labeling. This approach was shown to achieve state-of-the-art performance for all the NLP tasks investigated. In the field of image processing, LeCun et al. [1998] proposed a cheque reading system, based on handwritten digits recognition. In this system several tasks need to be performed: segmentation, feature extraction, single digit recognition and finally digits sequences recognition. Again, all these tasks are trained jointly, leveraging the deep learning approach. More recently, end-to-end approaches based on deep convolutional neural networks have been shown to yield state-of-the-art performance in object recognition [Krizhevsky et al., 2012, He et al., 2015]. Such an approach has also been successfully applied to deep reinforcement learning, yielding the first system to master the game of Go [Silver et al., 2016].

In speech recognition, acoustic models based on deep neural networks (DNNs) have received a lot of attention in recent years. These kind of networks are composed of many hidden layers. They are used in the framework of hybrid Hidden Markov Model/Artificial Neural Networks (HMM/ANN) [Bourlard and Morgan, 1994]. They have been shown to yield better systems than standard “shallow” neural networks [Hinton et al., 2012]. The first systems based on the DNN approach relied on the standard cepstral-based features. Recently, there has been growing interests in using “intermediate” representations, standing between raw signal and classical cepstral-based features, such as filterbank energies or magnitude spectrum. Overall, most of the ASR systems based on the deep neural network approach still rely on the “divide and conquer” approach, where the main task is divided into sub-tasks. The success stories of the end-to-end approach in other fields motivate us to ask: can we apply such approach to speech recognition?

The objective of this thesis is to investigate end-to-end trained systems for automatic speech recognition. Specifically, we investigate integrating each of the classical steps (features extraction, modeling and decoding), illustrated in Figure 1.1(a), in one single system, trained in an end-to-end manner using deep architectures. To this end, we take an incremental approach to the problem. First, we investigate an acoustic modeling approach that learns the relevant features and the classifier jointly, using the raw speech signal as input, illustrated in Figure 1.1(b). Next, we focus on end-to-end sequence recognition where the features, the classifier and the decoder are globally trained in a discriminative manner, illustrated in Figure 1.1(c).

1.2 Contributions

As mentioned above, this thesis follows two main axes of research. The first axis is devoted to joint learning of features and classifier for acoustic modeling using the temporal raw speech signal as input. The second axis of research focuses on end-to-end sequence modeling.

Along the first axis of research, in the framework of hybrid HMM/ANN based ASR, we develop a convolution neural networks (CNNs) based acoustic modeling approach that takes raw speech signal as input and estimates phone class conditional probabilities. We will show that using temporal raw speech as input to a CNN-based system leads to competitive systems on

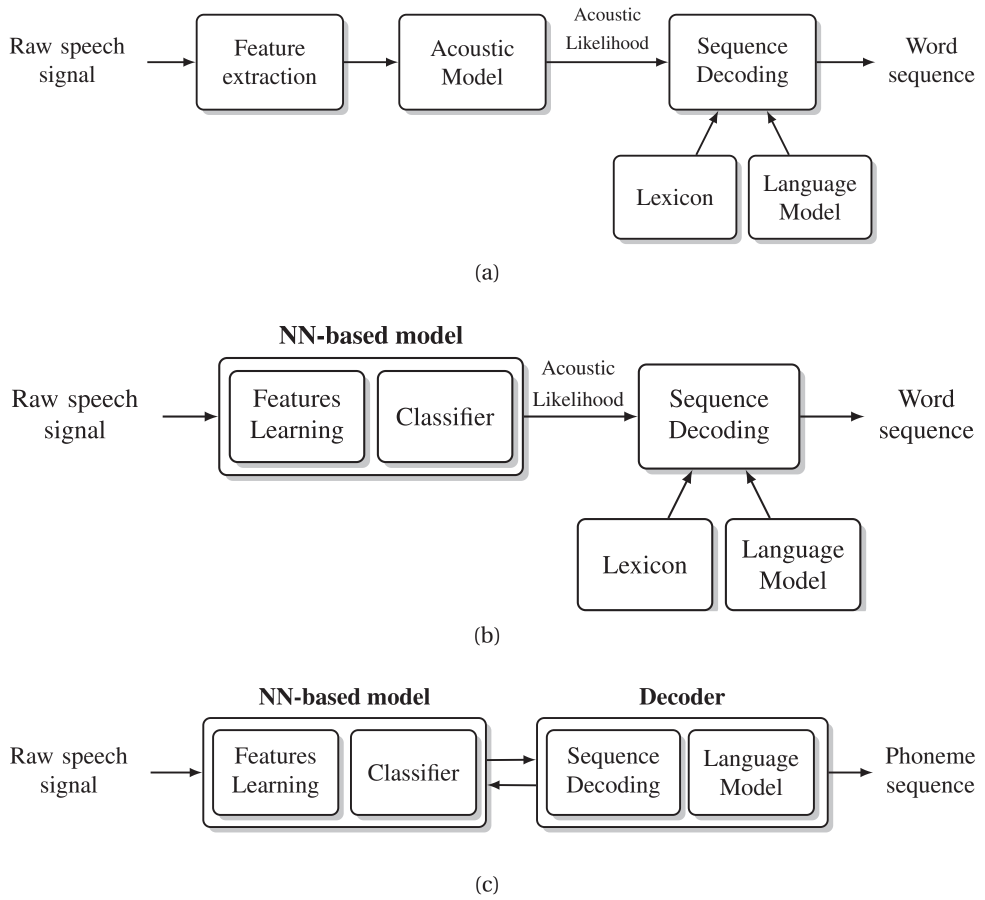

Raw speech signal
Feature extraction
Acoustic Model
Acoustic Likelihood
Sequence Decoding
Word sequence
Lexicon
Language Model
(a)
NN-based model
Raw speech signal
Features Learning
Classifier
Acoustic Likelihood
Sequence Decoding
Word sequence
Lexicon
Language Model
(b)
NN-based model
Decoder
Raw speech signal
Features Learning
Classifier
Sequence Decoding
Language Model
Phoneme sequence
(c)

Figure 1.1 – Illustration of the incremental approach: (a) standard system, (b) joint feature and classifier training and (c) end-to-end phoneme sequence recognition.

both phoneme recognition and continuous speech recognition task. This work was partly published in [Palaz et al., 2013a, 2015b].

In the proposed approach, the features are learned automatically with the classifier. Thus, two questions that arise are that what information is the neural network learning and how it is learning? We present an analysis of the network, and compare these findings against the classical approach of feature extraction. More specifically, we will show that:

- The first convolution acts as a filterbank, which (1) processes the signal at sub-segmental level (~ 2 ms) and (2) models the spectral envelope of the short-term signal. Specifically, in signal processing terms, a dictionary of matched filters is learned that capture formant-like information “in-parts”.
- The features learned by the CNNs have some level of invariance across domains and languages, and are more discriminative than the standard cepstral-based features

A part of this work was published in [Palaz et al., 2015a].

Building on the discriminative capabilities of the learned features, we present a study of the CNN-based system using deep features, i.e. many feature learning layers, and shallow classifier, i.e. simple linear classifier. We will show that this approach allows to reduce the number of parameters of the system while retaining the performance. A preliminary work in this direction was published in [Palaz et al., 2014a].

As the features are automatically learned from the signal, a natural question that arises is: are such systems robust to noise? To this aim, we propose a robust CNN approach, referred to as normalized CNN, which is based on online normalization of intermediate representations. We will show that the proposed CNN-based approach yields more robust systems when compared to conventional approach using cepstral features (with feature-level normalization). A preliminary investigation on noise robustness was published in [Palaz et al., 2015a].

The second axis of research focuses on end-to-end sequence-to-sequence conversion, where the relevant features, classifier and decoder are learned jointly. In that regard, we propose an end-to-end phoneme recognition system based on conditional random fields (CRF) that learns in a weakly-supervised manner phoneme segmentation and predicts phoneme sequence given raw speech as input. A part of this work was published in [Palaz et al., 2013b, 2014b].

Finally, we propose a weakly-supervised CNN-based approach that given a bag-of-word representation of utterances in the training set learns to locate and classify words. We demonstrate the potential of the approach through a keyword spotting study. A part of this work was published in [Palaz et al., 2016].

1.3 Organization of the Thesis

The remainder of the thesis is organized as follows:

- Chapter 2, Background, gives an overview of the standard ASR systems. A review on neural network-based acoustic models is then presented. Sequence-to-sequence conversion approach is then reviewed. An overview of keyword spotting systems is then presented.
- In Chapter 3, CNN-based ASR using Raw Speech Signal as Input, we present the CNN-based acoustic modeling approach, where the features are learned jointly with the classifier. We present in detail the proposed architecture and evaluate it on multiple tasks and languages.
- Chapter 4, Analysis of Proposed CNN-based System, presents the analysis of the feature learning stage of the CNN-based system and contrasts with the conventional short-term speech processing feature extraction.

- Chapter 5, Deep Features and Shallow Classifier, is devoted to the study of the CNN-based system using a linear classifier.
- Chapter 6, Towards Noise-Robust Raw Speech-based Systems, is devoted to the investigation of noise robustness of the CNN-based system on two benchmark corpora.
- In Chapter 7, End-to-end Phoneme Sequence Recognition, we present the CRF-based end-to-end sequence recognition approach where the features, the classifier and the decoder are trained jointly in an end-to-end manner for phoneme recognition.
- Chapter 8, Jointly Learning to Locate and Classify Words, is devoted to the weakly-supervised CNN-based approach using bag-of-words representations.
- Chapter 9, Conclusions, finally concludes the thesis along with possible directions for future research.

2 Background

In this chapter we provide a background on standard automatic speech recognition. We then present an overview on the recent advances in neural network-based acoustic modeling. A survey on the up-and-coming sequence-by-sequence conversion approach is then presented. Finally, an overview of the keyword spotting task is presented.

2.1 Overview

Automatic Speech Recognition (ASR) aims at converting a waveform signal S into a sequence of words W. In statistical terms, this problem can be formulated as finding the most likely word sequence given the input S:

$$W^* = \underset{W \in \mathcal{W}}{\operatorname{argmax}} P(W|S,\Theta), \tag{2.1}$$

where $\mathcal{W}$ denotes the set of hypotheses and Θ denotes the parameters. In the remainder of this chapter, Θ is dropped for the sake of clarity. To solve this problem, a general speech recognition system usually splits the task into three steps, as illustrated in Figure 2.1: feature extraction, acoustic modeling and sequence decoding.

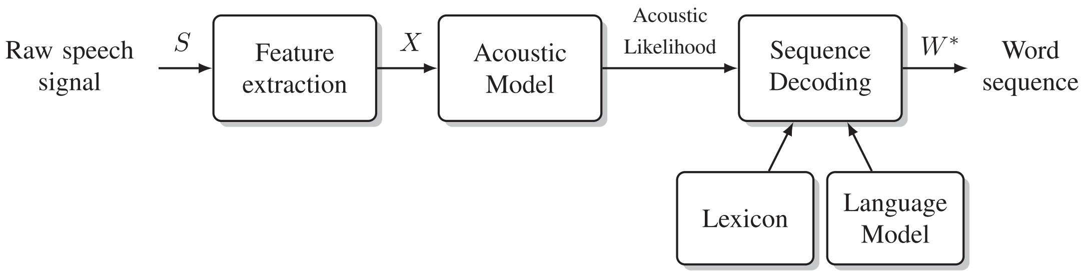

Raw speech signal
S
Feature extraction
X
Acoustic Model
Acoustic Likelihood
Sequence Decoding
W*
Word sequence
Lexicon
Language Model

Figure 2.1 – Overview of a general ASR system.

Feature extraction In this thesis, we express the speech signal $S = \{\mathbf{s}_1^c \dots \mathbf{s}_t^c \dots \mathbf{s}_T^c\}$ as a series of speech segments $\mathbf{s}_t^c = \{s_{t-c} \dots s_t \dots s_{t+c}\}$, composed of $2c$ speech samples s_t. In a first step, the waveform signal S is transformed into sequence of features or acoustic observation $X = [\mathbf{x}_1 \dots \mathbf{x}_t \dots \mathbf{x}_T]$, where $\mathbf{x}_t$ is a feature vector of dimension d representing the speech segment $\mathbf{s}_t^c$. The feature vector is usually obtained in two phases: an information selection phase, based on the task-specific knowledge of the phenomena and a dimensionality reduction phase. These two phases have been carefully hand-crafted, leading to state-of-the-art features such as cepstral-based features [Gold et al., 2011].

Acoustic modeling The acoustic modeling step typically models a statistical relationship between the features X and linguistically motived units, such as phonemes or phones.

Sequence decoding The sequence decoding step transcribes the input feature sequence X into a word sequence W. Broadly, this step can be expressed as:

$$\begin{aligned}
W^* &= \underset{W \in \mathcal{W}}{\operatorname{argmax}} \, P(W|X) && (2.2)\\
&= \underset{W \in \mathcal{W}}{\operatorname{argmax}} \, \frac{p(X|W)P(W)}{p(X)} && (2.3)\\
&\approx \underset{W \in \mathcal{W}}{\operatorname{argmax}} \, p(X|W)P(W) && (2.4)
\end{aligned}$$

where the Bayes rule is applied to in Equation (2.3) and the $P(X)$ is dropped in Equation (2.4) as it is independent of the word hypothesis and does not affect the maximization. In Equation (2.4), the acoustic likelihood of word hypothesis $p(X|W)$ is usually decomposed in sub-word unit acoustic likelihood through the **lexicon** and the *a priori* word hypothesis probability $P(W)$ is modeled by the **language model**.

2.2 Features

Speech signal is a non-stationary signal. Alternately, the statistical characteristics of the signal change over the time due to various reasons such as speech sound being produced, speaker variation, emotional state variation etc. In the case of ASR, we are primarily interested in the characteristic of the speech signal that relates to or differentiates the speech sounds.

Speech coding studies in telephony have shown that speech can be processed as short segments, transformed, transmitted and reconstructed while keeping the intelligibility or message intact [Rabiner and Schafer, 1978]. In particular, the studies have shown that short-term speech signal can be considered as output of a linear time invariant vocal tract filter excited by periodic or aperiodic vibration of vocal cords [Rabiner and Schafer, 1978]. Furthermore, speech intelligibility can be preserved by preserving the envelop structure of the short-term spectrum of speech signal, which characterizes the vocal tract system [Schroeder and Atal,

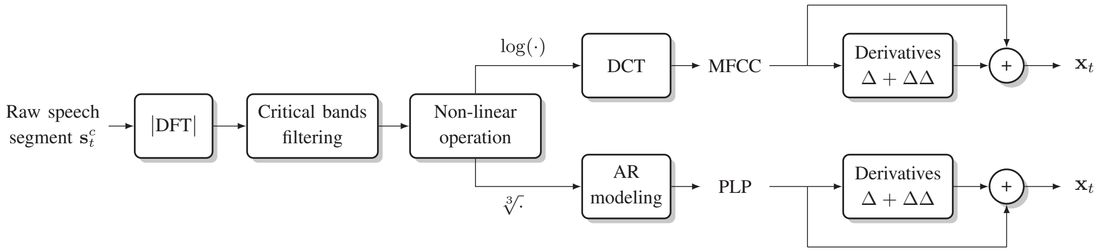


Figure 2.2 – MFCC and PLP extraction pipelines. |DFT| denotes the magnitude of the discrete Fourier transform, DCT denotes the magnitude of the discrete cosine transform, AR modeling stands for auto-regressive modeling, Δ and $\Delta\Delta$ denote the first and second order derivatives across time, respectively.

1985]. The two most common spectral-based features Mel frequency cepstral coefficient (MFCC) [Davis and Mermelstein, 1980] and perceptual linear prediction cepstral coefficient (PLP) [Hermansky, 1990] are built on those aspects while integrating the knowledge about speech and sound perception.

As illustrated in Figure 2.2, the extraction of MFCC or PLP feature involves: (1) transformation of short-term speech signal to frequency domain; (2) filtering the spectrum based on critical bands analysis, which is derived from speech perception knowledge; (3) applying a non-linear operation; and (4) applying a transformation to get reduced dimension decorrelated features. This process only models the local spectral level information on a short time window typically of 20-30 ms. The information about speech sound is spread over time. To model the temporal information intrinsic in the speech signal dynamic features are computed by taking approximate first and second derivative of the static features [Furui, 1986].

2.3 HMM-based Speech Recognition

State-of-the-art ASR systems are based on the Hidden Markov Model (HMM). An overview of this approach is presented below. The reader can refer to [Rabiner, 1989] for more details.

A hidden Markov model is a discrete model based on latent variable used to model temporal sequence. The features sequence $X = \{\mathbf{x}_1 \dots \mathbf{x}_t \dots \mathbf{x}_T\}$ is assumed to be generated by a sequence of *hidden* states $Q = \{q_1 \dots q_t \dots q_T\} \in \mathcal{Q}$. Each hidden state emits a observation from an emission probability distribution $p(\mathbf{x}_t|q_t)$, where the states are associated to a class $i \in \{1, \dots, I\}$. Formally, the HMM approach is based on Equation (2.4) which separates the task in two independent steps: the acoustic likelihood $p(X|W)$ estimation and the estimation of the prior language model probability $P(W)$.

2.3.1 Acoustic Likelihood Estimation

In the HMM framework, the acoustic likelihood $p(X|W)$ is estimated by:

$$
\begin{aligned}
p(X|W) &= \sum_{Q \in \mathcal{Q}} p(X, Q|W) && (2.5)\\
&= \sum_{Q \in \mathcal{Q}} p(X|Q, W)P(Q|W) && (2.6)\\
&= \sum_{Q \in \mathcal{Q}} p(X|Q)P(Q|W) && (2.7)\\
&\approx \max_{Q \in \mathcal{Q}} p(X|Q)P(Q|W) && (2.8)\\
&\approx \max_{Q \in \mathcal{Q}} \prod_{t=1}^{T} p_e(\mathbf{x}_t|q_t = i)P_{tr}(q_t = i|q_{t-1} = j) && (2.9)
\end{aligned}
$$

where the Bayes rules $p(X, Q|W) = p(X|Q, W)P(Q|W)$ is applied to Equation (2.6), it is assumed that the acoustic likelihood $p(X|Q, W)$ is independent of words given the state sequence in Equation (2.7) and where the Viterbi approximation, where the sum over all possible state sequence is replaced by the most probable state sequence is used in Equation (2.8). Equation (2.9) arises from the HMM assumptions, which are: (1) the acoustic observation $\mathbf{x}_t$ at time t depends only on the current state q_t, i.e. the observations are *i.i.d* and (2) the current state q_t depends only on the previous state q_{t-1}, following the first order Markovian assumption. $p_e(\mathbf{x}_t|q_t = i)$ are the emission probabilities for class i and $P_{tr}(q_t = i|q_{t-1} = j)$ are the transition probabilities between classes i and j at time t.

Two main approaches that are typically used to estimate the emission probabilities $p_e(\mathbf{x}_t|q_t = i)$ are Gaussian Mixture Model (GMM) and Artificial Neural Networks (ANN).

HMM/GMM Approach

In the HMM/GMM system, the emission probabilities are estimated by a mixture of Gaussian distributions:

$$
p_e(\mathbf{x}_t|q_t = i) = \sum_{j=1}^{J} c_{ij} N(\mathbf{x}_t, \mu_{ij}, \Sigma_{ij}), \tag{2.10}
$$

where J denotes the number of Gaussians, c_{ij} denote the weight for Gaussian distribution $\mathcal{N}(\mathbf{x}_t, \mu_{ij}, \Sigma_{ij})$.

$$
\mathcal{N}(\mathbf{x}_t, \mu_{ij}, \Sigma_{ij}) = \frac{1}{(2\pi)^{d/2}|\Sigma_{ij}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}_t - \mu_{ij})^T \Sigma_{ij}^{-1}(\mathbf{x}_t - \mu_{ij})\right) \tag{2.11}
$$

where d denotes the dimension of $\mathbf{x}_t$, μ_{ij} and Σ_{ij} denotes the mean vector and the covariance matrix, respectively.

Hybrid HMM/ANN Approach

The hybrid HMM/ANN proposed by Bourlard and Morgan [1994] is an ASR system based on the HMM approach, where the emission probabilities are estimated by artificial neural networks (ANN). An ANN is a discriminative classifier, described in details later in the Section 2.4.1. In this approach, the ANN estimates the class conditional probabilities $P(q_t = i|\mathbf{x}_t)$ for the feature frame $\mathbf{x}_t$, for each subword unit class $i \in \{1,\ldots,I\}$. The emission probabilities $p_e(\mathbf{x}_t|q_t = i)$ of the HMM states are scaled likelihoods which are obtained using the Bayes rule, by dividing the ANN output by the class prior probability $P(q_t = i)$,

$$p_e(\mathbf{x}_t|q_t = i) \propto \frac{p(\mathbf{x}_t|q_t = i)}{p(\mathbf{x}_t)} = \frac{P(q_t = i|\mathbf{x}_t)}{P(q_t = i)} \; \forall i \in \{1,\ldots,I\} \tag{2.12}$$

The prior state probability $P(q_t = i)$ is often estimated by counting on the training set. This framework will be used in this thesis in the first four chapters of this thesis.

It is worth mentioning that the feature input is usually composed of a feature vector $\mathbf{x}_t$ of the speech signal at time frame t and the feature vectors from preceding c time frames and following c time frames.

In both approaches, the parameters are learned by optimizing a maximum likelihood-based cost function using the expectation-maximization (EM) algorithm [Rabiner, 1989, Rabiner and Juang, 1993]. The two main approaches for estimating the parameters of HMM are the forward-backward algorithm or Baum-Welch algorithm [Baum et al., 1970, Rabiner, 1989] and the Viterbi training [Juang and Rabiner, 1990]. ANNs are typically trained in Viterbi EM framework, which consists of two iterative steps:

- Expectation (or E-step): Find the best state sequence given the current parameters.
- Maximization (or M-step): Train a new ANN with a cost function based on local classification error.

This process is illustrated in Figure 2.3. In this approach, at each M-step, a new neural network has to be trained from scratch, which requires each time several epochs of training. In that respect, this approach can be time consuming for large databases. Instead, the common approach is to train a HMM/GMM system to obtain a segmentation and then train an ANN afterwards.

2.3.2 Lexicon

Modeling the relations between all possible words and the acoustic observation is practically infeasible. Therefore, words are usually modeled as a sequence of subword units, given by the pronunciation lexicon. The most popular subword unit is the phoneme (or phone), the smallest unit in the phonology of languages [O'Shaughnessy, 1987]. The subword unit set

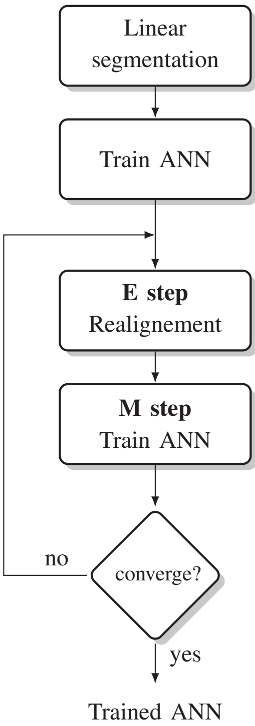


Figure 2.3 – Bloc diagram of the Viterbi EM approach.

can be context independent units or context-dependent units [Schwartz et al., 1985]. In the latter case, each context-independent unit in context with neighboring units is considered as a separate unit. For example the word '*that*' would be represented as "/dh/ /ae/ /t/" in the case of context-independent units and as "/dh+ae/ /dh-ae+t/ /ae-t/" in the case of context-dependent units. However, there are many unobserved context-dependent units during training. This issue is usually addressed by using clustering-based techniques [Young et al., 1994]. State-of-the-art ASR systems use context-dependent units. This information is modeled through the HMM states.

2.3.3 Language Model

$P(W)$ is estimated using a language model, which essentially models the transition between words. Formally, $P(W)$ can be estimated as:

$$P(W) = \prod_{m=1}^{M} P(w_m | w_1, w_2, \dots, w_{m-1}) \tag{2.13}$$

However, such estimation is a difficult problem, as the number of previous words is variable. Usually, n-gram statistical language models are used, where the probability of the current word depends only on the $n-1$ previous words:

$$P(W) = \prod_{m=1}^{M} P(w_m | w_{m-(n-1)}, \ldots, w_{m-1}) \tag{2.14}$$

Typically, bigram language model ($n = 2$) and trigram language model ($n = 3$) are used [Bahl et al., 1983, Jelinek, 1997]. Such models are usually estimated by counting on a large collection of text. To handle the problem of unobserved word combination, a smoothing approach is usually used, such as back-off, interpolation or discounting [Katz, 1987, Kneser and Ney, 1995]. Recently, more advanced language model based on recurrent neural network have also been proposed [Mikolov et al., 2010, Lecorvé and Motlicek, 2012].

2.3.4 Decoding and Evaluation

During decoding, the acoustic likelihood estimation $p(X|W)$ and the language model $P(W)$ are combined to infer the most probable word sequence. The Viterbi algorithm [Forney, 1973] is used to find the most probable word sequence. A full breadth search is however infeasible in practice, therefore pruning using beam search techniques [Greer et al., 1982] is usually used to efficiently infer the word sequence.

The performance of ASR systems is evaluated in term of phoneme error rate (PER) for studies on phoneme sequence recognition and on word error rate (WER) for studies on continuous speech recognition. These two metrics are computed using the Levenstein distance, a dynamic programming algorithm, between the ground truth sequence and the recognized sequence, expressed in percentage:

$$\text{PER/WER} = \frac{Del + Sub + Ins}{N} \cdot 100 \ \ [\%] \tag{2.15}$$

where N denotes the total number of phoneme or word occurrence in the ground truth, Del denotes the number of deletions, Sub denote the number of substitution and Ins the number of insertions. The performance can be also expressed in term of word recognition rate (WRR), i.e. $100 - \text{WER}$ [%].

2.4 Neural Networks-based Acoustic Modeling

In this section, we formally define the artificial neural networks framework and then present an overview of the ANN-based acoustic modeling in speech recognition.

2.4.1 Artificial Neural Networks

Artificial neural networks are non-linear adaptive models which model the relationship between an vector input $\mathbf{x}$ of dimension d_x and a vector output $\mathbf{y}$ of dimension d_y. Historically, neural networks were inspired by biological systems [McCulloch and Pitts, 1943]. The first attempt was the perceptron introduced by Rosenblatt [1958] as a linear classifier. It was then extend to non-linear classification and shown to be an universal approximator [Cybenko, 1989, Hornik et al., 1989].

In this thesis, we use the following framework. A neural network is composed of several *layers*, each layer being a specific operation. The simplest architecture of a neural network is composed of a linear layer, a matrix vector product, and a non linear transfer function. It can be expressed as:

$$\mathbf{y} = h(M\mathbf{x} + \mathbf{b}) \tag{2.16}$$

where M, the weight matrix of dimension $d_x \times d_y$ and $\mathbf{b}$ the bias vector of dimension d_x are the parameters of the model, and $h(\cdot)$ is a non-linear transfer function, such as hyperbolic tangent or sigmoid.

The most common architecture is composed of one or more *hidden* layers. It is often referred to as multilayer perceptron (MLP). One hidden layer MLP can be written as:

$$\mathbf{y}_h = h(M_1\mathbf{x} + \mathbf{b}_1) \tag{2.17}$$

$$\mathbf{y}_{out} = h(f(\mathbf{x})) = h(M_2\mathbf{y}_h + \mathbf{b}_2) \tag{2.18}$$

where $\mathbf{y}_h$ denotes the hidden representation or output of the hidden layer and $f(\mathbf{x})$ denotes the network output. The parameters of the model are the weight matrix and bias vector of each layer. The number of hidden units is a *hyper-parameter*, which has to be selected empirically.

Neural networks can be used for both classification and regression. In this thesis, we use them for classification. In classification task, neural networks model the relationship between an input $\mathbf{x}$ and a target, or class, label $i \in \{1, \ldots, I\}$. The network output $f(\mathbf{x})$ is thus a vector of size I, where each component $f_i(\mathbf{x})$ represents a score for each class. To compute the posterior probability $P(i|\mathbf{x})$, a softmax non-linearity can be used on the output scores of the network [Bridle, 1990b]:

$$P(i|\mathbf{x}) = \frac{e^{f_i(\mathbf{x})}}{\sum_j e^{f_j(\mathbf{x})}} \tag{2.19}$$

In literature, it has been shown that neural networks can estimate the posterior probabilities when trained using the cross-entropy or squared-error criteria [Richard and Lippmann, 1991, Morgan and Bourlard, 1995], as presented below.

Given a training set of N examples and their respective labels $(\mathbf{x}_n, i_n)$, $n = 1, \ldots, N$, the neural network can be trained by optimizing a cost function $\mathcal{L}$ (also called objective function, or criterion). The typical cost function for pattern classification is the cross-entropy criterion, based on a proximity measure between the network output and the “one hot” representation of the class i, i.e. a vector of size I with 1 for the ith component and 0 elsewhere. Formally, it can be expressed as:

$$\mathcal{L}(\theta) = \sum_{n=1}^{N} \log(P(i_n|\mathbf{x}_n,\theta)) \tag{2.20}$$

where the log-probability is computed as:

$$\log(P(i|\mathbf{x},\theta)) = f_i(\mathbf{x},\theta) - \underset{j}{\text{logadd}}(f_j(\mathbf{x},\theta)) \tag{2.21}$$

and the logadd operation is defined as:

$$\underset{j}{\text{logadd}}(z_j) = \log(\sum_j e^{z_j}). \tag{2.22}$$

The back-propagation algorithm [Rumelhart et al., 1985, LeCun, 1989] is used to train the model. This algorithm consists of propagating the error backward in the network using the chain derivative rule. The parameters θ are updated by making a gradient step:

$$\theta \longleftarrow \theta + \lambda \frac{\partial \log(P(i|\mathbf{x},\theta))}{\partial \theta} \tag{2.23}$$

where λ denotes the learning rate. The parameters are usually initialized randomly and can be updated either by batch, i.e. by accumulating the cost gradient from several examples, or by using the stochastic gradient descent technique [Bottou, 1991] which randomly iterates over the training set, estimating the gradient of the likelihood for one example between each update. The main issue when training neural networks models is overfitting, which is the tendency of the network to learn the training set “by heart”, thus decreasing its generalization capabilities. To prevent that, a validation set is often used during training. At each iteration, the model predictions on the validation set are evaluated, and the training is stopped when the validation set classification accuracy decreases [Morgan and Bourlard, 1989]. This method is referred to as early stopping. The validation set can also be used for selecting the hyper-parameters, i.e. selecting the best model.

2.4.2 Architectures

Neural networks-based systems have gained a lot of interest since mid 1980’s in speech recognition for acoustic modeling. The first successful applications were obtained on phoneme recognition [Hinton and Lang, 1988, Waibel et al., 1989]. Later, it was extended to isolated word recognition [Bottou et al., 1989]. For continuous speech recognition, successful results have also been obtained [Haffner, 1992], but on small vocabularies. At the same time, the hybrid HMM/ANN approach [Bourlard and Wellekens, 1990, Bengio, 1993, Renals et al., 1994, Morgan and Bourlard, 1995] was developed.

In the remainder of this section, we present the recent architectures developed for NN-based acoustic modeling. A survey can be found in [Hinton et al., 2012].

Deep Neural Network

Following the success of the hybrid HMM/ANN system, the recent increase in computing resources have led to the development of Deep Neural Networks (DNNs). These types of neural networks are composed of several hidden layers:

$$\mathbf{y}_{out} = h(M_n h(M_{n-1} \dots h(M_1 \mathbf{x}))) \tag{2.24}$$

where M_n denotes the weight matrix of layer n and $h(\cdot)$ denotes the activation function. This approach has been shown to improve performance in speech recognition tasks compared to standard MLP with one hidden layer [Hinton et al., 2012]. However, these types of networks are known to be difficult to train [Larochelle et al., 2009, Glorot and Bengio, 2010], specially when the amount of data is limited. To address this issue, pre-training techniques have been developed. These techniques are based on learning "good" intermediate representation, usually using unsupervised generative models. These representations then serve as starting point for discriminative training. Approaches such as the greedy layer-wise training [Bengio et al., 2007] or the noisy auto-encoder approach [Vincent et al., 2008] have been proposed. In the speech community, one of the most popular technique is the deep belief networks [Hinton et al., 2006] approach. This pretraining approach is based on the restricted Boltzmann machines framework and aims at maximizing the likelihood of the joint probability of data and labels. Other regularization techniques have also been proposed, such as the dropout approach [Srivastava, 2013]. This technique is based on randomly setting to zero a certain amount of the weights at each update during training. The effect of this approach is to force the neurons to not rely on each other, thus improving the generalization capabilities of the network.

In literature, hybrid HMM/DNN systems have been proposed using standard cepstral-based features as input for phone recognition [Mohamed et al., 2009] and continuous speech recognition [Seide et al., 2011, Mohamed et al., 2011, Dahl et al., 2012]. Extracting bottleneck features have also been proposed [Yu and Seltzer, 2011, Sainath et al., 2012]. Using dropout has also been investigated for continuous speech recognition [Dahl et al., 2013]. More recently, there has been a growing interest in using "intermediate" representations (standing between waveform signal and classical features such as cepstral-based features) as input. Spectral-based features have been investigated for phoneme recognition task [Lee et al., 2009] and continuous speech recognition task [Mohamed et al., 2012, Bocchieri and Dimitriadis, 2013, Zhang et al., 2014]. Learning features from spectrum has been proposed in [Sainath et al., 2013c]. Learning feature from the raw speech signal has also been proposed [Jaitly and Hinton, 2011].

Convolutional Neural Network

Inspired by studies on visual cortex, a convolutional neural network (CNN) is the architecture of choice when dealing with sequential data [LeCun, 1989]. Instead of applying a linear transformation on a fixed-side input vectors, the CNN assumes that the input is a sequence of vector, and then a convolution of a chosen length applies a linear transformation. It means that for each input vector, the same transformation is applied to a window around the input. The output of the network can be seen as a higher-level representation of the input. One can stack convolution layers, to obtain a more abstract representation. This kind of network seems to be suited for speech, because the data is often represented as frame, and the surrounding frames of the input (the context) carry information related to the task.

In speech recognition community, this kind of networks has been referred to as Time-Delay Neural Network. They were initially studied on phoneme recognition using Mel-scale log filterbank energies as input [Waibel et al., 1989] and on isolated word recognition using Bark-scale log filterbank energies as input [Bottou et al., 1989]. In the recent years, using CNNs with filterbank energies as input has regained interest on phoneme recognition [Abdel-Hamid et al., 2012], continuous speech recognition [Deng et al., 2013, Sainath et al., 2013b,a] and distant speech recognition [Swietojanski et al., 2014]. Speaker adaptation technique has also been investigated in [Abdel-Hamid and Jiang, 2013].

Recurrent Neural Network

Recurrent Neural Networks (RNN) [Elman, 1990] are a class of neural networks in which connections between the units (or neurons) form a directed graph. In other words, to classify an example at a given time, a RNN-based model can access the predictions of the earlier examples. Bi-directional RNN [Schuster and Paliwal, 1997] have also been proposed, which are composed of two RNNs, one in each direction. Thus, the prediction at a given time can access predictions in both direction.

In the context of HMM-based speech recognition, recurrent neural networks based systems have been proposed. The alpha-net [Bridle, 1990a] is a RNN-based approach presented in the HMM framework. Robinson [1994] also proposed a recurrent network, where the output of the network is computed according to the present input and a hidden state variable, which depends on all the previous inputs. The main limitation of these models is the vanishing gradient problem, which limits their access to long range context. The Long Short Term Memory networks (LSTM) [Hochreiter and Schmidhuber, 1997] have been shown to address this issue. They are discussed in the next section.

Convolutional Neural Networks will be pivotal to this thesis, and will be formally dealt with in the remainder of the thesis.

2.5 Sequence-to-sequence Conversion

Speech recognition is in essence a sequence-to-sequence conversion problem, more specifically predicting a word sequence given an input speech signal (a sequence of numbers). As presented in the previous section, the HMM/ANN approach solves the problem into two steps: (1) each input frame is modeled by a local estimation of the acoustic likelihood and (2) the sequence is decoded using a language model. Each step is optimized independently.

Alternate approaches based on sequence-to-sequence modeling have been proposed. These approaches tend to estimate $P(W|X)$ in a more global manner. This is an emerging topic in speech recognition. In this section, we review two approaches: Long Short Term Memory (LSTM) and Conditional Random Fields (CRF).

2.5.1 Long Short Term Memory

Long Short Term Memory (LSTM) [Hochreiter and Schmidhuber, 1997] is a particular type of recurrent neural network composed of LSTM gates. A LSTM gate is able to learn which information to store or to delete. These gates can replace neuron units or be used in addition to them. Therefore, use of LSTM gates allows the network to have access to a very long context to model an input at a given time. The bi-directional LSTM (BLSTM) approach based on bi-directional RNN allows the network to have access to input context in both direction. This approach has been mainly studied for phoneme recognition [Graves and Schmidhuber, 2005, Graves et al., 2013]. Preliminary studies on continuous speech recognition were recently presented [Graves and Jaitly, 2014]. LSTM layers were also combined with other types of neural networks, such as CNNs, in the context of hybrid HMM/ANN framework [Deng and Platt, 2014, Sainath et al., 2015a].

2.5.2 Conditional Random Fields

The Conditional Random Fields (CRF), proposed by Lafferty et al. [2001], is a discriminative probabilistic model for segmenting and labelling sequential data. It is defined as a directed graphical model whose nodes are divided into two sets: the input sequence X and the label sequence Y. In this model, the conditional relationship $P(Y|X)$ is modeled. Formally, the CRF model is defined as a graph $G = (E, V)$, where E denotes the edges and V the vertices (or nodes). The conditional relationship is defined as:

$$P(Y|X) = \frac{1}{Z} \exp\left(\sum_{e \in E,k} a_k f_k(e, Y|_e, X) + \sum_{v \in V,k} b_k g_k(v, Y|_v, X) \right) \tag{2.25}$$

where $f(\cdot)$ and $g(\cdot)$ are fixed features, a and b are their respective weights and Z the normalization factor.

This model can be applied to phoneme recognition task, for example with $f(\cdot)$ representing phone classes scores and $g(\cdot)$ representing phone classes transition. In literature, this model has been investigated on phoneme recognition using MLP posteriors and phonological attribute as features [Morris and Fosler-Lussier, 2008]. It was later extended to an approach where the CRF backpropogates its error to the MLP-based classifier [Prabhavalkar and Fosler-Lussier, 2010]. Use of DNN-based classifier has also been proposed [Mohamed et al., 2010, Kubo et al., 2012]. This framework will be used in Chapter 7.

2.6 Keyword Spotting

The keyword spotting (KWS) problem consists of detecting a query in a spoken document. The query can be text-based or spoken. In the latter case, the task is called as query-by-example. This thesis focuses on the text-based query inputs. Formally, the KWS problem can be formulated as a statistical hypothesis testing problem:

$$\frac{p(X|H_1)}{p(X|H_0)} > \Delta \tag{2.26}$$

where X denotes a sequence of acoustic features from the spoken document, H_1 is the hypothesis denoting the presence of the query term and H_0 is a hypothesis denoting the absence of the query term, $p(X|H_j)$ is the likelihood of hypothesis H_j and Δ the detection threshold.

In order to estimate the ratio in Equation (2.26), state-of-the-art KWS systems employ a few or all components of HMM-based ASR system. In literature, different KWS approaches have been proposed, which are discussed briefly below. This task will be used in Chapter 8.

2.6.1 Approaches

Acoustic Matching

In this approach, the system uses the trained acoustic model and lexicon of a existing ASR system [Rohlicek et al., 1989, Rose and Paul, 1990, Wilpon et al., 1990, Bourlard et al., 1994a, Szöke et al., 2005]. The query terms is therefore modeled as a sequence of sub-word unit form the lexicon. Usually a sequence model is built where the query term is preceded and followed by a “filler” HMM, which models a non-query term, typically a phone loop HMM. The likelihood is then estimated using Viterbi algorithm, and then compared to a background sequence model that does not contain the query term, and a decision is made based on the ratio of likelihood. Another approach, instead of using the background likelihood, is to obtain the first and last frame of the query term from the best path, and to estimate a confidence score for the segment [Bernardis and Bourlard, 1998, Williams and Renals, 1999].

Lattice Search

One of the simplest way to detect query term is to transcribe the spoken data using an ASR system, and perform a text search. But the system is then prone to the errors committed by the ASR system. A way to remedy this problem is to perform a search using word-based lattices generated by ASR system [Odell, 1995] instead of a single best output [Saraclar and Sproat, 2004, Can and Saraclar, 2011]. Phoneme-based lattice generation has also been proposed [Yu and Seide, 2004, Szöke et al., 2005]. The main advantage of this approach is that the lattices can be stored to perform multiple query searches.

Discriminative Approach

Recently, a discriminative KWS approach based on Support Vector Machine (SVM) was proposed in [Keshet et al., 2009]. The KWS system is trained discriminatively in an end-to-end manner by optimizing the area under the Receiver Operating Characteristic curve. This approach has been found to outperform the acoustic matching approach and has the advantage of using minimal resources of ASR system [Keshet et al., 2001].

2.6.2 Metric

Keyword spotting is a detection task, which consists of detecting all occurrences of a given keyword in the spoken document. In other words, a KWS system can be seen as a binary detection system for each utterance. Such systems can make two kind of mistakes: false alarm and missed detection. To evaluate the performance, these two types or errors have to be considered. The standard metric for binary detection task is the F measure or F1 score [Fawcett, 2006], which combines precision and recall. For keyword spotting, two metrics are often used: the Receiver Operating Characteristic (ROC) [Fawcett, 2006] curve and Maximum Term Weighted Value (MTWV).

ROC Curve

The Receiver Operating Characteristic [Fawcett, 2006] is often used to evaluate binary decision processes. It consists of a plot of the true positive rate (TPR) against the false positive rate (FPR) obtained by varying the detection threshold. To compare systems, the Area Under Curve (AUC) [Fawcett, 2006] is derived from the ROC. Higher the AUC, better is the system. AUC=1 means perfect detection. In keyword spotting, these metrics face the problem of normalization (needed for computing the rates), as there is no clear definition of trials. To remedy this issue, the National Institute of Standards and Technology (NIST) has proposed another metric, referred to as Maximum Term Weighted Value (MTWV).

MTWV

The Term Weighted Value (TWV) metric was proposed by NIST during the 2006 STD pilot evaluation [Fiscus et al., 2007]. It measures one minus the average value lost by the system. The maximum possible value is 1, indicating a perfect output. An empty output yields a TWV of 0. Negative value are also possible. Formally, TWV is expressed as:

$$TWV(\Delta) = 1 - average\{P_{miss}(term, \Delta) + \beta P_{FA}(term, \Delta)\} \tag{2.27}$$

for a given threshold Δ, with P_{miss} denote the missed detection probabilities and P_{FA} the false alarm probabilities. They are computed as:

$$P_{miss}(term, \Delta) = 1 - \frac{N_{correct}(term, \Delta)}{N_{true}(term)}, \tag{2.28}$$

$$P_{FA}(term, \Delta) = \frac{N_{spurious}(term, \Delta)}{N_{NT}(term)}, \tag{2.29}$$

where for a given term, $N_{correct}$ is the number of correct detections, $N_{spurious}$ is the number of incorrect detection, N_{true} is the true number of occurrence and N_{NT} is the number of opportunities for incorrect detection. It is estimated as $N_{NT} = n_{pr} * T_{speech} - N_{true}$, where n_{pr} is the number of trials per second, and T_{speech} is the total length of the test data in seconds. The weight β is computed as:

$$\beta = \frac{C}{V}(Pr_{term}^{-1} - 1) \tag{2.30}$$

where $\frac{C}{V}$ denote cost over value ratio and Pr_{term} the prior probability of a term. In order to perform a comparison between KWS systems, the Maximum TWV is often used. It is simply defined as the maximum of the Term Weighted Value:

$$MTWV = \max_{\Delta} TWV(\Delta) \tag{2.31}$$

2.7 Summary

In this chapter, we provided a brief overview of speech recognition systems, including the HMM/GMM-based system and the hybrid HMM/ANN system. We then presented a literature overview on NN-based acoustic modeling, sequence-to-sequence conversion and keyword spotting.

3 CNN-based ASR using Raw Speech Signal as Input

In speech recognition, the standard acoustic modeling mechanism can be seen as a process of applying transformations guided by prior knowledge about speech production and perception on the speech signal, and subsequent modeling of the resulting features by a statistical classifier. More recently, inspired by the success of deep learning approaches in the field of text processing and vision towards building end-to-end systems [Collobert et al., 2011b, He et al., 2015] as well as by the success of DNNs in ASR, researchers have started questioning the intermediate step of feature extraction. In that direction, several studies have been carried where filterbank or critical band energies estimated from the short-term signal instead of cepstral features are used as input of convolutional neural networks based systems [Abdel-Hamid et al., 2012, Sainath et al., 2013b, Swietojanski et al., 2014] or short-term magnitude spectrum is used as input to DNN proposed [Mohamed et al., 2012, Lee et al., 2009]. Figure 3.1 illustrates a case where, instead of transforming the critical band energies into cepstral features, the critical band energies and its derivatives are fed as input to the ANN.

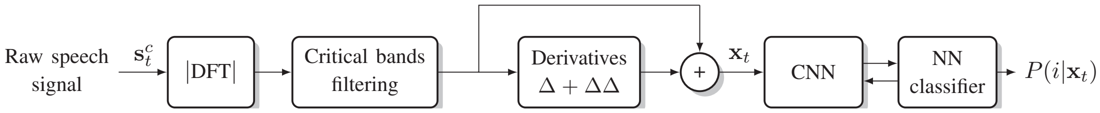


Figure 3.1 – Typical CNN-based system using Mel filterbanks coefficient as input [Swietojanski et al., 2014].

In this chapter, we go one step further and propose a novel approach where the features and the classifier are jointly learned. Alternately, in this approach the raw speech signal is input to an ANN that classifies speech sounds. During training the neural network automatically learns both the relevant features and the classifier. The output of the trained neural network is then used as emission probabilities of HMM states as done in hybrid HMM/ANN approach. Such an approach can not only be motivated by recent advances in machine learning but also from previous works in the speech literature in which direct modeling of raw speech signal has been proposed for speech recognition.

In the remainder of this chapter, we present a brief survey of related literature. We then present the proposed CNN-based approach and the recognition studies.

3.1 Related Work

The first initiative towards directly modeling the raw speech signal was inspired by speech production model, i.e. an observed speech signal can be seen as an output of a time varying filter excited by a time varying source. Specifically, one of the first theoretical work in that direction by Portiz [Poritz, 1982] was inspired by linear prediction technique which can deconvolve the excitation source and the vocal tract system through time domain processing. Poritz's work was later revisited as switching autoregressive HMM [Ephraim and Roberts, 2005], and more recently in the framework of switching linear dynamical systems [Mesot and Barber, 2008]. These techniques were investigated in an isolated word recognition setup where word-based models are trained. It was found that in comparison to HMM-based ASR system using cepstral features these approaches yield performance comparable under clean conditions and significantly better performance under noisy conditions [Mesot and Barber, 2008]. In [Sheikhzadeh and Deng, 1994], an approach to model raw speech signal was proposed using auto-regressive HMM. In this approach, each sample of the speech signal is the observation as opposed to a vector of speech samples in the approach proposed in [Poritz, 1982]. Each state models the observed speech sample as a linear combination of past samples plus a "driving sequence" (assumed to be a Gaussian *i.i.d* process). The potential of the approach was demonstrated on classification of speaker-dependent discrete utterances consisting of 18 highly confusable stop consonant-vowel syllables. These works demonstrated the potential of modeling directly the raw speech signal. However, their gain compared to conventional cepstral-based features is not clear, and they were never studied on large scale task such as continuous speech recognition.

More recently, using raw speech signal as input to discriminative systems has been investigated. Combination of raw speech and cepstral features in the framework of support vector machine has been investigated for noisy phoneme classification [Yousafzai et al., 2009]. Features learning from raw speech using neural networks-based systems has been investigated in [Jaitly and Hinton, 2011]. In this approach, the learned features are post-processed by adding their temporal derivatives and used as input for another neural network. Thus, this approach still follows the “divide and conquer” approach. In comparison to that, in our approach, the features are learned jointly with the acoustic model in an end-to-end manner. There are other more recent works that have followed the proposed approach. We discuss them later in Section 4.3.

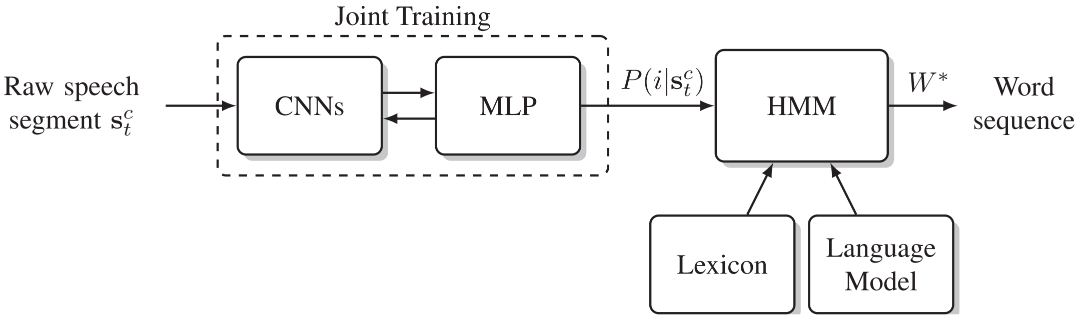


Figure 3.2 – Overview of the proposed CNN-based approach.

3.2 Proposed CNN-based Approach

We propose a novel acoustic modeling approach based on convolutional neural networks (CNN), where the input speech signal $\mathbf{s}_t^c = \{s_{t-c} \ldots s_t \ldots s_{t+c}\}$ is a segment of the raw speech signal taken in context of c milliseconds. The input signal is processed by several convolution layers and the resulting intermediate representations are classified to estimate $P(i|\mathbf{s}_t^c)$, $\forall i$, as illustrated in Figure 3.2. $P(i|\mathbf{s}_t^c)$ is subsequently used to estimate emission scaled-likelihood $p_e(\mathbf{s}_t^c|i)$ as per Equation (2.12). As presented in Figure 3.3, the network architecture is composed of several filter stages, followed by a classification stage. A filter stage involves a convolutional layer, followed by a temporal pooling layer and a non-linearity, $HardTanh(\cdot)$. The number of filter stages is determined during training. The feature stage and the classifier stage are jointly trained using the back-propagation algorithm.

The proposed approach employs the following understanding:

1. Speech is a non-stationary signal. Thus, it needs to be processed in short-term manner. Traditionally, in the literature guided by Fourier spectral theory and speech analysis-synthesis studies the short-term window size is set as 20-40 ms. The proposed approach follows the general idea of short-term processing. However, the size of the short-term window is a hyper-parameter which is automatically determined during training.
2. Feature extraction is a filtering operation. This can be simply observed from the fact that generic operations such as Fourier transform, discrete cosine transform etc. are filtering operations. In conventional speech processing, the filtering takes place in both frequency (e.g. filter-bank operation) and time (e.g. temporal derivative estimation). The convolution layers in the proposed approach build on these understandings. However, aspects such as the number of filter-banks and their parameters are automatically learned during training.
3. Though the speech signal is processed in short-term manner, the information about the speech sounds is spread across time. In conventional approach, the information spread across time is modeled by estimating temporal derivatives and by using contextual

information, i.e. by appending features from preceding and following frames, at the classifier input. In the proposed approach the intermediate representations feeding into the classifier stage are estimated using long time span of input speech signal, which is again determined during training.

In essence the proposed approach with minimal assumptions or prior knowledge learns to process the speech signal to estimate $P(i|\mathbf{s}_t^c)$.

3.2.1 Convolutional Neural Networks

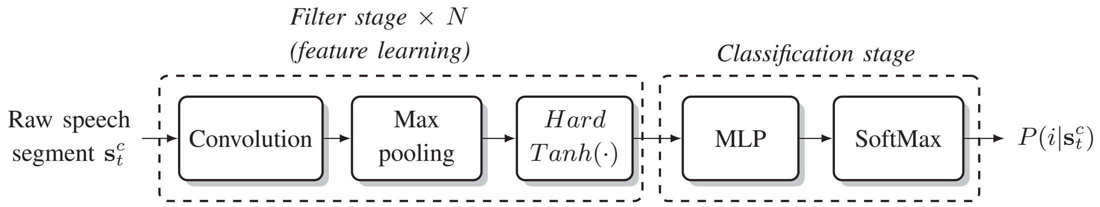


Figure 3.3 – Overview of the convolutional neural network architecture. Several stages of convolution/pooling/tanh might be considered. Our network included 3 stages. The classification stage can have multiple hidden layers.

Convolutional Layer

While “classical” linear layers in standard MLPs accept a fixed-size input vector, a convolution layer is assumed to be fed with a sequence of T vectors/frames $\{\mathbf{y}_1 \ldots \mathbf{y}_t \ \ldots \ \mathbf{y}_T\}$. In this work, $\mathbf{y}_t$ is either a segment of input raw speech $\mathbf{s}_t^c$ (for the first convolution layer) or a intermediate representation output by the previous convolution layers. A convolutional layer applies the same linear transformation over each successive (or interspaced by dW frames) windows of kW frames, as illustrated in Figure 3.4. The transformation at frame t is formally written as:

$$M \begin{pmatrix} \mathbf{y}_{t-(kW-1)/2} \\ \vdots \\ \mathbf{y}_{t+(kW-1)/2} \end{pmatrix}, \tag{3.1}$$

where M is a $d_{out} \times d_{in}$ matrix of parameters, d_{in} denotes the input dimension and d_{out} denotes the dimension of the output frame. In other words, d_{out} filters (rows of the matrix M) are applied to the input sequence.

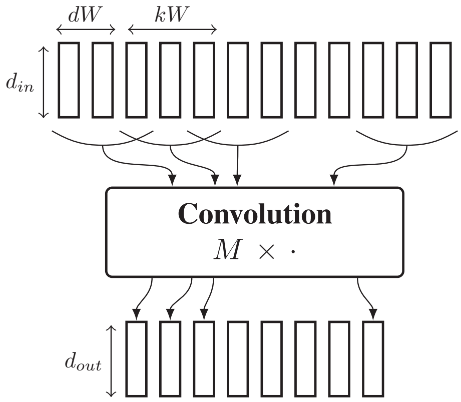


Figure 3.4 – Illustration of a convolutional layer. d_{in} and d_{out} are the dimension of the input and output frames. kW is the kernel width (here $kW = 3$) and dW is the shift between two linear applications (here, $dW = 2$).
.

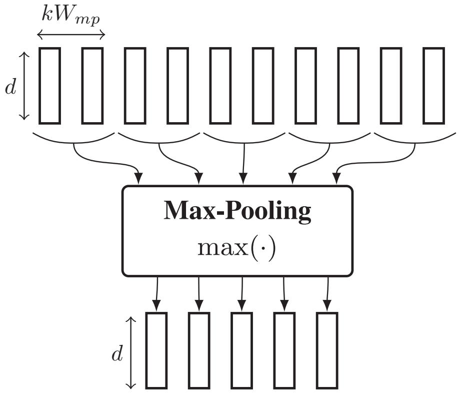


Figure 3.5 – Illustration of max-pooling layer. kW_{mp} is the number of frame taken for each max operation (here, $kW_{mp} = 2$) and d represents the dimension of input/output frames (which are equal). In this case, the shift $dW_{mp} = kW_{mp}$.

Max-pooling Layer

These kind of layers perform local temporal max operations over an input sequence, as shown in Figure 3.5. More formally, the transformation at frame t is written as:

$$\max_{t-(kW_{mp}-1)/2 \leq k \leq t+(kW_{mp}-1)/2} \mathbf{y}_k[d] \qquad \forall d \tag{3.2}$$

with $\mathbf{y}$ being the vector/frames input and d the dimension. These layers increase the robustness of the network to minor temporal distortions in the input.

SoftMax Layer

The $Softmax$ [Bridle, 1990b] layer interprets network output scores $f_i(\mathbf{s}_t^c)$ of an input $\mathbf{s}_t^c$ as conditional probabilities, for each class label i:

$$P(i|\mathbf{s}_t^c) = \frac{e^{f_i(\mathbf{s}_t^c)}}{\sum_j e^{f_j(\mathbf{s}_t^c)}} \tag{3.3}$$

Non-linearity

This kind of layer applies a non-linearity to the input. In this work, we use the $HardTanh$ layer, defined as:

$$HardTanh(x) = \begin{cases} -1 & \text{if } x < -1 \\ x & \text{if } -1 \leq x \leq 1 \\ 1 & \text{if } x > 1 \end{cases} \tag{3.4}$$

3.2.2 Network Training

The network parameters θ are learned by maximizing the log-likelihood $\mathcal{L}$, given by:

$$\mathcal{L}(\theta) = \sum_{t} \log(P(i|\mathbf{s}_t^c, \theta)) \tag{3.5}$$

for each input $\mathbf{s}_t^c$ and its corresponding label i, over the whole training set, with respect to the parameters of each layer of the network, as presented in Section 2.4.1. Optimizing this likelihood is performed using the stochastic gradient descent algorithm [Bottou, 1991].

3.2.3 Illustration of a Trained Network

In the proposed approach, in addition to the number of hidden units in each hidden layer of the classification stage, the filter stage has number of hyper-parameters, namely, time span of input speech signal w_{in} used to estimate $P(i|\mathbf{s}_t^c)$ (here, $c = \frac{w_{in}}{2}$), number of convolution layers, kernel or temporal window width kW at input of each convolution layer, dW shift of the temporal window at the input of each convolution layer, max pooling kernel width kW_{mp} and shift dW_{mp}. In the present work, all of these hyper-parameters are determined during training based on frame-level classification accuracy on validation data.

Figure 3.6 illustrates the trained feature stage of the proposed CNN approach on TIMIT corpus. The details of the training can be found in the following Section 3.3. The filter stage has three convolution layers and it takes a window of 250 ms speech signal w_{in} as input to estimate $P(i|\mathbf{s}_t^c)$ every 10 ms. The figure also illustrates the temporal information κ modeled by the output of each layer and the temporal shift δ. Briefly, the first convolution layer models in a fine grain manner the changes in the signal characteristics over time, i.e. processes 1.8 ms of speech ($kW = 30$ samples) every 0.6ms ($dW = 10$ samples). The subsequent convolution layers then filter and temporally integrate the output of the first convolution layer to yield an intermediate feature representation that is input to the classifier stage, which eventually yields an estimate of $P(i|\mathbf{s}_t^c)$

It is worth pointing out that the dimensionality of the intermediate representation at the feature learning stage output depends upon the number of convolution stages and the max-pooling kernel width. As it can be seen that max-pooling is done without temporal overlap. So

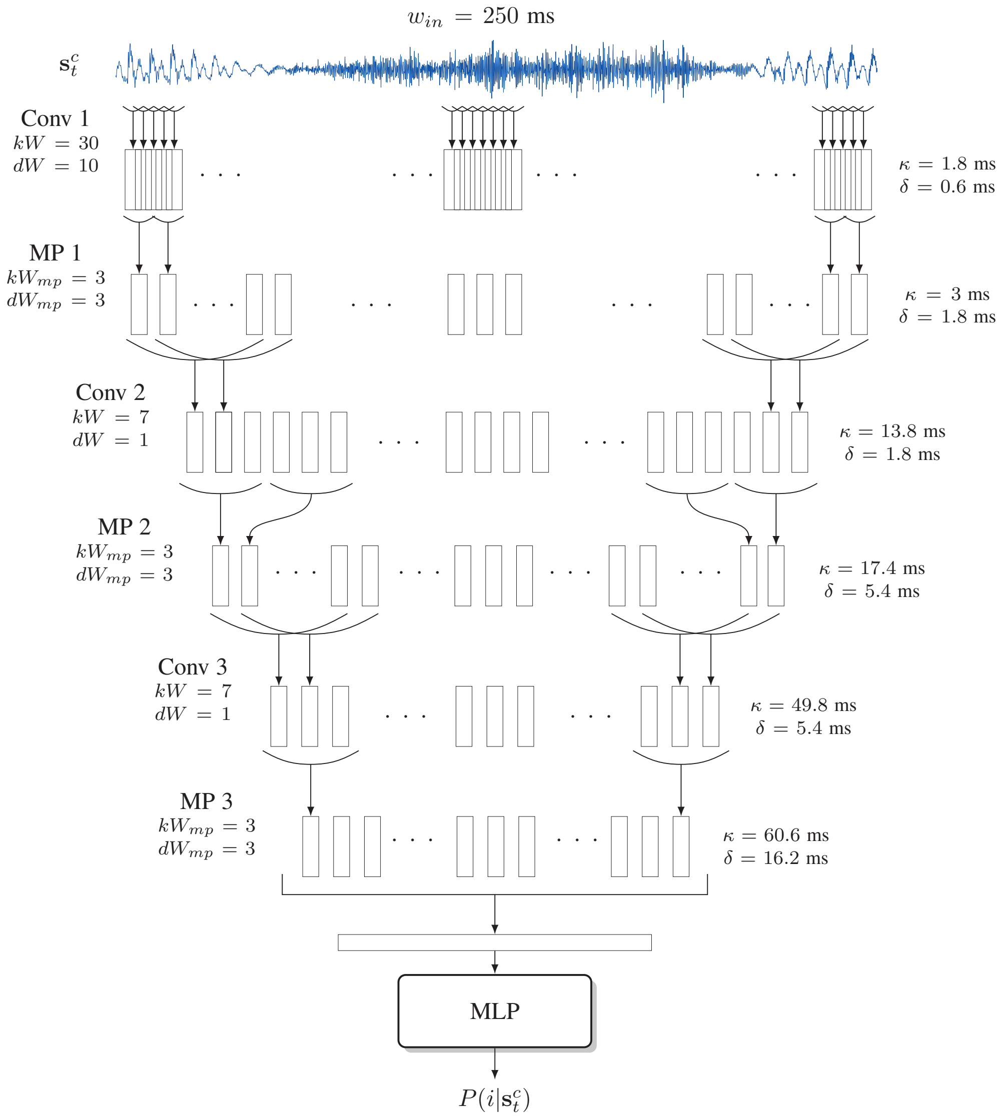

$w_{in} = 250$ ms
$\mathbf{s}_t^c$
Conv 1
$kW = 30$
$dW = 10$
$\kappa = 1.8$ ms
$\delta = 0.6$ ms
MP 1
$kW_{mp} = 3$
$dW_{mp} = 3$
$\kappa = 3$ ms
$\delta = 1.8$ ms
Conv 2
$kW = 7$
$dW = 1$
$\kappa = 13.8$ ms
$\delta = 1.8$ ms
MP 2
$kW_{mp} = 3$
$dW_{mp} = 3$
$\kappa = 17.4$ ms
$\delta = 5.4$ ms
Conv 3
$kW = 7$
$dW = 1$
$\kappa = 49.8$ ms
$\delta = 5.4$ ms
MP 3
$kW_{mp} = 3$
$dW_{mp} = 3$
$\kappa = 60.6$ ms
$\delta = 16.2$ ms
MLP
$P(i|\mathbf{s}_t^c)$

Figure 3.6 – Illustration of the feature stage of CNN trained on TIMIT to classify 183 phoneme classes. κ and δ indicates the temporal information modeled by the layer and the shift respectively. Non-linearity layers are applied after each max-pooling.

at each convolution stage, in addition to filtering minor temporal distortions, max-pooling operation acts as a down sampler.

3.3 Recognition Studies

In this section, we present automatic speech recognition studies to show the potential of the proposed approach. We compare it against the conventional approach of spectral-based feature extraction followed by ANN training on different tasks and languages, namely, (a)

TIMIT phoneme recognition task, (b) Wall street journal (WSJ) 5k task, (c) Swiss French Mediaparl task and (d) Swiss German Mediaparl task. The objective of these studies is to demonstrate the viability of the proposed approach by comparing it against standard MFCC features for estimating phoneme class posterior probability.

The remainder of the section is organized as follows. Section 3.3.1 presents the different datasets and setup used for the studies. Section 3.3.2 presents the different systems that are trained and evaluated. Section 3.3.3 presents the results of the recognition studies.

3.3.1 Databases and Setup

TIMIT

The TIMIT acoustic-phonetic corpus [Garofolo et al., 1993] consists of 3,696 training utterances (sampled at 16kHz) from 462 speakers, excluding the SA sentences. The cross-validation set consists of 400 utterances from 50 speakers. The core test set is used to report the results. It contains 192 utterances from 24 speakers. Experiments were performed using 61 phoneme labels, with three states, for a total of 183 targets as in [Mohamed et al., 2009]. After decoding, the 61 hand labeled phonetic symbols are mapped to 39 phonemes, as presented in [Lee and Hon, 1989].

Wall Street Journal

The Wall Street Journal (WSJ) corpus is an English corpus based on read microphone speech. The SI-284 set of the corpus [Woodland et al., 1994] is formed by combining data from WSJ0 and WSJ1 databases. The set contains 36416 sequences sampled at 16 kHz, representing around 80 hours of speech. Ten percent of the set was taken as validation set. The Nov'92 set was selected as test set. It contains 330 sequences from 10 speakers. The dictionary was based on the CMU phoneme set, 40 context-independent phonemes (including silence). We obtained 2776 clustered context-dependent (cCD) units, i.e. tied-states, by training a context-dependent HMM/GMM system with decision tree based state tying. We used the bigram language model provided with the corpus. The test vocabulary contains 5000 words.

Mediaparl

MediaParl is a bilingual corpus [Imseng et al., 2012] containing data (debates) in both Swiss German and Swiss French which were recorded at the Valais parliament in Switzerland. Valais is a state which has both French and German speakers with high variability in local accents specially among German speakers. Therefore, MediaParl provides a real-speech corpus that is suitable for ASR studies. In our experiments, audio recordings with 16 kHz sampling rate are used.

The Swiss German part of the database, referred to as *MP-DE*, is partitioned into 5955 sequences from 73 speakers for training (14 hours), 876 sequences from 8 speakers for validation (2 hours) and and 1692 sequences from 7 speakers (4 hours) for test. 1101 tied-states were used in the experiments, following the best system available on this corpus [Razavi et al., 2014]. The vocabulary size is 16,755 words. The dictionary is provided in SAMPA format with a phone set of size 57 (including sil) and contains all the words in the train, development and test set. A bigram language model was used.

The Swiss French part of the database, referred to as *MP-FR*, is partitioned into 5471 sequences from 107 speakers for training (14 hours) , 646 sequences from 9 speakers for validation (2 hours) and and 925 sequences from 7 speakers (4 hours) for test. 1084 tied-states were used in the experiments, as presented in [Razavi and Magimai.-Doss, 2014]. The vocabulary size is 12,035 words. The dictionary is provided in SAMPA format with a phone set of size 38 (including sil) and contains all the words in the train, development and test set. A bigram language model was used.

3.3.2 Systems

In this section, for each task studied, we present the details of the conventional spectral feature based baseline systems and the proposed CNN-based system using raw speech signal as input. All neural networks were initialized randomly and trained using the Torch7 toolbox [Collobert et al., 2011a]. The HTK toolbox [Young et al., 2002] was used for the HMMs and the cepstral features extraction.

Conventional Cepstral Feature based System

On each task, we have two baseline hybrid HMM/ANN systems which differ in terms of ANN architecture. More precisely, one hidden layer MLP (denoted as ANN-1H) based system and three hidden layer MLP (denoted as ANN-3H) based system. These ANNs estimate $P(i|\mathbf{x}_t$ where $\mathbf{x}_t$ is a cepstral feature vector. The details of the baseline systems for the different tasks are as follows,

- TIMIT: We treat the one hidden layer MLP based system and the three hidden layer MLP based system without pre-training i.e. random initialization reported in [Mohamed et al., 2012, Figure 6] as the baseline systems. Our motivation in doing so is that they are one of the best cepstral feature based systems reported in the literature on this task. In these systems, the input to the MLPs were 39 dimensional MFCC features ($c_0 - c_{12} + \Delta + \Delta\Delta$) with five frames preceding and five frames following context (i.e. input dimension 39 × 11). ANN-1H has 2048 nodes in the hidden layer and ANN-3H has 1024 nodes in each of the three hidden layers.
- WSJ: We trained an ANN-1H and an ANN-3H to classify 2776 tied-states. The input to the MLP was 39 dimensional MFCC features ($c_0 - c_{12} + \Delta + \Delta\Delta$) with four frames preceding

and four frames following context (i.e. input dimension 39 × 9). The MFCC features are computed using a frame size of 25ms and a frame shift of 10 ms. ANN-1H had 1000 nodes in the hidden layer and ANN-3H had 1000 nodes in each hidden layer.

- MP-DE: We use the setup of the best performing hybrid HMM/ANN using a three hidden layers MLP classifying 1101 clustered context-dependent units reported in [Razavi et al., 2014] as the baseline ANN-3H system. The ANN has 1000 nodes in each hidden layer. We trained an ANN-1H with 1000 hidden units for the present study. The inputs to the ANNs were 39 PLP cepstral features ($c_0 - c_{12} + \Delta + \Delta\Delta$) with four frames preceding and four frames following context. The frame size and frame shift were 25 ms and 10ms, respectively.

- MP-FR: We use the setup of the best performing hybrid HMM/ANN using a three hidden layers MLP classifying 1084 clustered context-dependent units reported in [Razavi and Magimai.-Doss, 2014] as the baseline ANN-3H system. The ANN has 1000 nodes in each hidden layer. We trained an ANN-1H with 1000 hidden units for the present study. The inputs to the ANNs were 39 PLP cepstral features ($c_0 - c_{12} + \Delta + \Delta\Delta$) with four frames preceding and four frames following context. The frame size and frame shift were 25 ms and 10ms, respectively.

Proposed CNN-based System

We trained the proposed CNN-based $P(i|\mathbf{s}_t^c)$ estimator using raw speech signal. The inputs are simply composed of a window of the speech signal (hence $d_{in} = 1$, for the first convolutional layer). The utterances are normalized such that they have zero mean and unit variance, which is in line with the literature [Sheikhzadeh and Deng, 1994]. No further pre-processing is performed. The hyper-parameters of the network are: the time span of the input signal (w_{in}), the kernel width kW and shift dW of the convolutions, the number of filters d_{out}, maxpooling width KW_{mp} and shift dW_{mp} and the number of nodes in the hidden layer(s). Note that the input d_{in} for the first convolution layer is one (i.e. a sample of the speech signal). For the remaining layers, the d_{in} is the product of d_{out} of the previous layer and kW of that layer. These hyper parameters were determined by early stopping on the validation set, based on frame classification accuracy. The ranges which were considered for a coarse grid search are reported in Table 3.1. We used the TIMIT task to narrow down the hyper-parameters search space, as it provided fast turn around experiments.

For each of the tasks, we trained CNNs with one hidden layer (denoted as CNN-1H) and three hidden layers (denoted as CNN-3H) similar to the different MLP architectures in the baseline systems. We found that three convolution layers consistently yields the best validation accuracy across all the tasks. The CNN architecture found for each of the task is presented in Table 3.2. The shift of max-pooling kernel $dW_{mp} = 3$ was found for all the layers on all the tasks. As we will observe later, the capacity of the CNN-based approach in terms of number of parameters lies at the classifier stage. So, for fair comparison with the baseline systems,

Table 3.1 – Range of hyper parameters considered for the grid search.

Parameters	Units	Range
Input window size (w_{in})	ms	100-700
Kernel width of the first conv. (kW_1)	samples	10-90
Kernel width of the n^{th} conv. (kW_n)	frames	1-11
Number of filters per kernel (d_{out})	filters	20-100
Max-pooling kernel width (kW_{mp})	frames	2-6
Number of hidden units in the classifier	units	200-1500

we restricted the search for the number of hidden nodes in the hidden layer(s) such that the number of parameters are comparable to the respective baseline systems. The output classes were same as the case of cepstral feature-based system, i.e. for TIMIT task 183 phone classes, for WSJ task 2776 cCD units, for MP-DE task 1101 cCD units and for MP-FR task 1084 cCD units.

Table 3.2 – Architecture of CNN-based system for different tasks. HL=1 denotes CNN-1H and HL=3 denotes CNN-3H. w_{in} is expressed in terms of milliseconds. The hyper-parameters kW, dW, d_{out} and kW_{mp} for each convolution layer is comma separated. HU denotes the number of hidden units. 3 × 1000 means 1000 hidden units per hidden layer.

	HL	w_{in}	kW	dW	d_{out}	kW_{mp}	HU
TIMIT	1	250	30,7,7	10,1,1	80,60,60	3,3,3	1000
	3	250	30,7,7	10,1,1	80,60,60	3,3,3	3x1000
WSJ	1	210	30,7,7	10,1,1	80,60,60	3,3,3	1000
	3	310	30,7,7	10,1,1	80,60,60	3,3,3	3x1000
MP-DE	1	210	30,7,7	10,1,1	80,60,60	3,3,3	1000
	3	310	30,7,7	10,1,1	80,60,60	3,3,3	3x1000
MP-FR	1	190	30,7,7	10,1,1	80,60,60	3,3,3	1000
	3	310	30,7,7	10,1,1	80,60,60	3,3,3	3x1000

3.3.3 Results

In this section we present the results of the studies on different tasks. For the sake of completeness, for the speech recognition studies we also report performance on HMM/GMM system. For MP-DE and MP-FR, the best performing HMM/GMM systems reported in [Razavi et al., 2014] and [Razavi and Magimai.-Doss, 2014], respectively are presented. It is worth pointing out that they have more number of tied states than the hybrid HMM/ANN and the CNN-based system presented here.

TIMIT

Table 3.3 presents the results on TIMIT phone recognition task in terms of phoneme error rate (PER). It can be observed that the proposed CNN-based approach outperforms the conventional cepstral feature based system. In [Mohamed et al., 2012, Figure 6], ANNs with different hidden layers were investigated with cepstral feature as input. The best performance of 23.0% PER for the case of random initialization is achieved with 7 hidden layers, 3072 hidden nodes per layer and 17 frames temporal context (8 preceding and 8 following) 23.0% PER. With pre-training, the best performance of 22.3% is achieved with 6 hidden layers, 3072 hidden nodes per layer and 17 frames temporal context. The CNN-3H system performs better than those systems as well.

Table 3.3 – Phoneme error rate of different systems on the core test set of the TIMIT corpus.

Input	System	#Conv. params.	#Class. params.	PER (in %)
MFCC	ANN-1H [Mohamed et al., 2012]	na	1.2M	24.5
MFCC	ANN-3H [Mohamed et al., 2012]	na	2.6M	22.6
RAW	CNN-1H	63k	920k	22.8
RAW	CNN-3H	52k	2.9M	21.9

Table 3.4 contrasts our results with a few prominent results on TIMIT using ANNs. Inputs of these systems are either MFCCs (computed as presented in Section 3.3.2), Mel filterbanks energies (abbreviated FBANKs) or “improved” MFCC features (denoted MFCC+LDA+MLLT+fMLLR), which are obtained by applying decorrelation processes (linear discriminant analysis and maximum likelihood linear transform) and speaker normalization (feature-space maximum likelihood linear regression) [Rath et al., 2013] to the original MFCC coefficient. One can see that the proposed approach outperforms most of the systems using MFCCs features. Systems using improved MFCCs features yields better results than the proposed approach, mainly due to the speaker normalization technique, which could be developed for the proposed approach. Finally, one can see that RNN-based systems (the three last entries of Table 3.4) clearly yield the best performance.

WSJ

The results for the CSR study on the WSJ corpus in presented in Table 3.5. for the baseline systems and the proposed system. As it can be observed, the CNN-1H based system outperforms the ANN-1H based baseline system, and the CNN-3H based system also outperforms the ANN-3H based system, with as many parameters.

Table 3.4 – Phoneme error rate of different systems reported in literature on the core test set of the TIMIT corpus.

Method (input)	PER (in %)
Augmented CRFs (MFCC) [Hifny and Renals, 2009]	26.6
HMM/DNNs 6 layers (MFCC) [Mohamed et al., 2012]	22.3
Deep segmental NN (MFCC) [Abdel-Hamid et al., 2013]	21.9
Proposed approach	**21.9**
HMM/DNNs 6 layers (MFCC+LDA+MLLT+fMLLR) [Lu et al., 2016]	18.5
CTC transducers (FBANKs) [Graves et al., 2013]	17.7
Attention-based RNN (FBANKs) [Chorowski et al., 2015]	17.6
Segmental RNN (MFCC+LDA+MLLT+fMLLR) [Lu et al., 2016]	17.3

Table 3.5 – Word Error Rate on the Nov'92 testset of the WSJ corpus

Input	System	#Conv. params.	#Class. params.	WER (in %)
MFCC	GMM	na	4M	5.1
MFCC	ANN-1H	na	3.1M	7.0
MFCC	ANN-3H	na	5.6M	6.4
RAW	CNN-1H	46k	3.1M	6.7
RAW	CNN-3H	61k	5.6M	5.6

MP-DE

The results on the Mediaparl German corpus are presented in Table 3.6. The CNN-1H based system outperforms the GMM-based system, the ANN-1H based system and the ANN-3H system with four times less parameters. The CNN-3H system yields the best performance.

Table 3.6 – Word Error Rate on the testset of the MP-DE corpus.

Input	System	#Conv. params.	#Class. params.	WER (in %)
PLP	GMM [Razavi et al., 2014]	na	3.8M	26.6
PLP	ANN-1H	na	2.2M	26.7
PLP	ANN-3H [Razavi et al., 2014]	na	8.8M	25.5
RAW	CNN-1H	61k	1.6M	24.4
RAW	CNN-3H	92k	8.7M	23.5

MP-FR

The results on the Mediaparl French corpus are presented in Table 3.7. Again, a similar trend can be observed, i.e. the CNN-1H based system outperforms the ANN-1H baseline and the CNN-3H outperforms the ANN-3H based system.

Table 3.7 – Word Error Rate on the testset of the MP-FR corpus.

Input	System	#Conv. params.	#Class. params.	WER (in %)
PLP	GMM [Razavi and Magimai.-Doss, 2014]	na	3.8M	26.8
PLP	ANN-1H	na	2.2M	27.0
PLP	ANN-3H [Razavi and Magimai.-Doss, 2014]	na	8.8M	25.5
RAW	CNN-1H	61k	1.5M	25.9
RAW	CNN-3H	92k	8.7M	23.9

3.4 Summary

In this chapter, we proposed a novel CNN-based acoustic modeling approach that automatically learns relevant representations from the speech signal and estimates phone class conditional probabilities for ASR. Our studies showed that with minimal assumptions the proposed approach is able to learn to process the speech signal to estimate phone class conditional probabilities $P(i|\mathbf{s}_t^c)$ and yield a system that outperforms conventional cepstral feature based system using ANN with multiple hidden layers. Furthermore, we consistently observed that the CNN-1H system yields performance comparable to ANN-3H system with considerably fewer number of parameters.

4 Analysis of Proposed CNN-based System

In the previous chapter, it was shown that the CNN-based approach using raw speech as input yields an ASR system that performs better than the system based on conventional approach with considerably less number of parameters. Thus, a question that arise is: what information is the neural network learning and how it is learning? Since the features are learned along with the classifier automatically from the data, yet another question that arises is: are these features domain or language dependent? To understand these aspects we first present an analysis of the system that gives insight about the information that is learned by the filters at the first convolution layer (Section 4.1) We then focus the analysis at the output of feature learning stage, where we evaluate the cross-domain and cross-lingual capabilities of the learned features (Section 4.2). The analyses are done using the corpora and the systems presented in the previous chapter.

4.1 First Convolution Layer

In this section, we present an analysis of the first convolution layer. We first provide an input level analysis, where the hyper-parameters of the layer (found experimentally) are compared against the conventional speech processing approach. We then show that the convolution layer can be interpreted as a bank of matching filters. Finally, we analyze how these filters respond to various inputs and present a method to understand the filtering process.

4.1.1 Input level Analysis

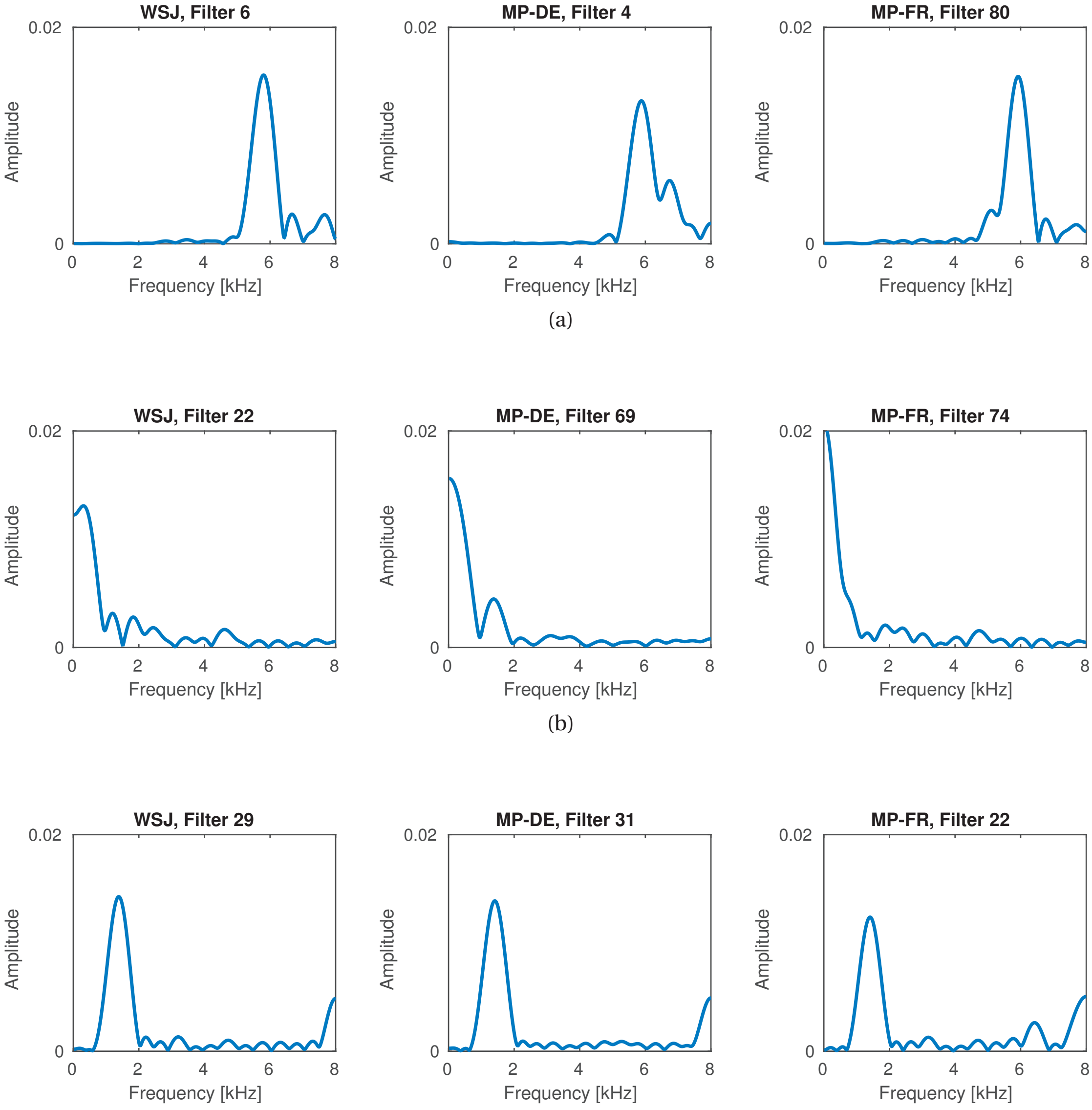
To learn to process raw speech signal and estimate $P(i|\mathbf{s}_t^c)$ the proposed approach employs many hyper-parameters which are decided based on validation data. We can get insight into the approach by relating or contrasting a few of the hyper-parameters to the traditional speech processing. First among that is time span of the signal w_{in} used to estimate $P(i|\mathbf{s}_t^c)$. From Table 3.2, we can observe that w_{in} varies from 190 ms - 310 ms. This is consistent with the literature which supports the idea of processing syllable length speech signal (around 200 ms) for classification of phones [Hermansky, 1998]. This aspect can be also observed in another way. Usually, in hybrid HMM/ANN system the input is the cepstral features (static + Δ + $\Delta\Delta$) at the current time frame and features of four preceding frames and four following frames. If the frame shift is 10 ms and the temporal derivatives are computed using two frames preceding and two frames following context then the 9 frame feature input models about 170 ms of speech signal.

Next we can understand how the speech signal of time span of 190 ms - 310 ms is processed at the input of the network through the kernel width (kW) and kernel shift (dW) of the first convolution stage. We can see from Table 3.2 that for all tasks kW is 30 speech samples and dW is 10 speech samples. Given that the sampling frequency is 16 kHz, this translates into a window of 1.8 ms and shift of about 0.6 ms. This is contrary to the conventional speech processing where typically the window size is about 25 ms, the shift is about 10 ms and the resulting features are concatenated at the classifier input. Note that in our case w_{in} is shifted by 10ms, however with in the window of 190 ms - 310 ms the speech is processed at sub-segmental level at the first convolution layer and subsequently processed by later convolution layers with different temporal resolutions to estimate $P(i|\mathbf{s}_t^c)$.

Such a sub-segmental processing at the first convolution layer could possibly be reasoned through signal stationarity assumptions. More precisely, the convolution filters at the first stage are learned by discriminating the phone classes at the output of the CNN. So, for the output of the convolution filter to be informative (for phone classification), the filter has to operate on stationary segments of the speech signal spanned by w_{in}. It can be argued that such a stationary assumption would clearly hold for one glottal cycle or pitch period of the speech signal. In such a case suppose if the limit of the observed pitch frequency is assumed to be 500 Hz, i.e. beyond adult speakers pitch frequency range, then a window size of 2 ms or less would ensure that the filters operate where the vocal tract system can be considered stationary i.e. with in a glottal cycle. This line of argument is also consistent with traditional feature extraction methods which tend to model the smooth envelope of the short-term spectrum, i.e. information related to vocal tract response, with quasi-stationarity assumptions.

4.1.2 Learned Filters

The first convolution layer learns a set of filters that operates on the speech signal in a similar way to filter bank analysis during MFCC or PLP cepstral feature extraction. In the case of MFCC or PLP cepstral feature extraction the number of filter banks and their characteristics are determined a priori using speech perception knowledge. For instance, the filters are placed either on Mel scale or on Bark scale. Furthermore, each of the filters cover only a part of the bandwidth, out of which the response is strictly zero. The number of filters are chosen based on bandwidth information. For instance, in the case of Mel scale around 24 filters for 4 kHz bandwidth (narrow band speech) and 40 filters for 8 kHz bandwidth (wide band speech) are typically used. While in the case of Bark scale, there are 15 filters for 4 kHz bandwidth and 19



(a)

(b)

(c)

Figure 4.1 – Examples of three close pairs of filters learned. The left column is from CNN-1H WSJ, the center one is from CNN-1H MP-DE, the right one is from CNN-1H MP-FR.

filters for 8 kHz bandwidth (see e.g. [Hönig et al., 2005]).

In contrast, in the proposed approach the number of filters and their responses are automatically learned in data-driven manner, i.e., while learning to estimate $P(i|\mathbf{s}_t^c)$. It can be observed from Table 3.2 that the number of filters for all the tasks is 80. This is well above the range typically used in speech processing. In order to understand the learned filter characteristics, we analyzed the filters learned on WSJ, MP-DE and MP-FR task in the following manner:

(i) The complex Fourier transform $\mathcal{F}$ of the filters learned on the WSJ, MP-DE and MP-FR tasks for CNN-1H case are computed using 1024 point FFT. The 512 point magnitude spectrum $|\mathcal{F}_m|$ of each filter m is then normalized, i.e. converted into a probability mass function. F_m denotes the normalized magnitude spectrum of filter m.

(ii) For each filter $m = 1, \cdots 80$ learned on WSJ, we find the closest filter $n = 1, \cdots 80$ learned on MP-DE and MP-FR using symmetric Kullback-Leibler divergence,

$$d(F_m, F_n) = \frac{1}{2} \cdot [D_{KL}(F_m \,||\, F_n) + D_{KL}(F_n \,||\, F_m)], \tag{4.1}$$

$$D_{KL}(F_m || F_n) = \sum_{u=1}^{512} F_m[u] \ln \frac{F_m[u]}{F_n[u]}, \tag{4.2}$$

where $F_m[u]$ is the normalized magnitude at u^{th} point of FFT of filter m of WSJ CNN-1H and $F_n[u]$ is the normalized magnitude at u^{th} point of FFT of filter n of MP-DE CNN-1H or MP-FR CNN-1H.

Figure 4.1 presents normalized frequency responses of a few filters learned on WSJ (on the left column) and the closest filters learned on the MP-DE task (on the middle column) and on the MP-FR task (on the right column). We can make two observations. First, the filters are focussing on different parts of the spectrum. However, unlike the filter banks in the MFCC or PLP cepstral feature extraction, the frequency response of the filters cover the whole bandwidth. Second, it can be observed that similar filters can be found across domain and languages, although there is a difference in the spectral balance, especially as observed in the case of Figure 4.1(b).

To further understand the characteristics of the learned filters, we estimated the cumulative frequency response of all the filters in the filterbank:

$$F_{cum} = \sum_{n=1}^{80} F_n \tag{4.3}$$

Figure 4.2 presents the gain normalized cumulative frequency responses for CNN-1H WSJ, CNN-1H MP-DE and CNN-1H MP-FR. We can make two key observations,

(i) Though the filters are learned on different languages and corpora, we can see that below

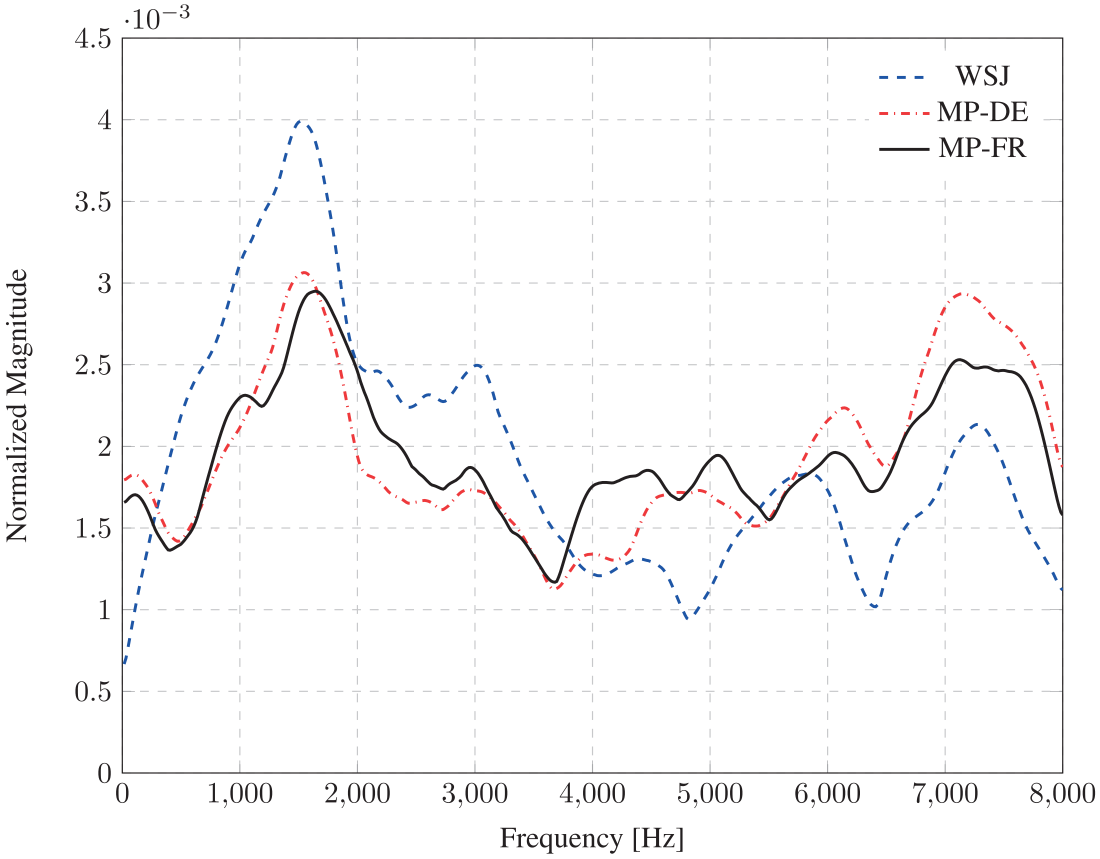


Figure 4.2 – Cumulative frequency responses of the learned filterbank on WSJ, MP-DE and MP-FR.

4000 Hz and above 6500 Hz the shape of frequency response for WSJ, MP-DE and MP-FR are similar. As the filters are operating on sub-segmental speech, we speculate that the peaks (high energy regions) are more related to the resonances in the vocal tract or phoneme discriminative invariant information. Between 4000 Hz and 6500 Hz, we can see that MP-DE and MP-FR have responses that closely match, but are different than WSJ. Overall we observe that the spectral balance for WSJ is different than for MP-DE and MP-FR. We attribute this balance mismatch mainly to the fact that the WSJ and the Mediaparl corpora are different domains in terms of type of speech (read vs. spontaneous) and recording environment (controlled vs real world). In the following sub-section and Section 4.2.2 we touch upon this aspect again.

(ii) Auditory filterbanks such as Mel scale filterbanks or Bark scale filterbanks are usually designed to have a cumulative frequency response that is flat. In other words, constant Q bandpass filterbank. In contrast to that, it can be seen that the cumulative frequency response of the learned filters is not constant Q bandpass. The main reason for that is standard filterbanks emerged from human sound perception studies considering the complete auditory frequency range or the bandwidth, so as to aid analysis and synthesis (reconstruction) of the audio signal. However, in our case these filters are learned for the purpose of discriminating phones, and the speech signal contains information other than just phones. The figure suggests that, for discriminating only phones, constant Q bandpass filterbank is not a necessary condition.

4.1.3 Response of Filters to Input Speech Signal

In Section 4.1.1, we observed that the speech signal of time span 190 ms - 310 ms is processed in sub-segmental manner. In the previous section, we observed that the filters that operate on sub-segment of speech signal are tuned to different parts of the spectrum during training. In other words, matched to different parts of the spectrum relevant for phone discrimination. In this section, we ascertain that by analyzing the response of the filters to the the input speech signal in relationship with phones.

The CNNs in the WSJ, MP-DE and MP-FR studies were trained to classify cCD units, which can be quite distinctive across languages. So, in order to facilitate the analysis across languages, we trained CNNs with single hidden layer on WSJ, MP-DE and MP-FR data to classify context-independent phones with same hyper parameters. We denote these CNNs as CNN-1H-mono WSJ, CNN-1H-mono MP-DE and CNN-1H-mono MP-FR, respectively.

As a first step, we analyzed the energy output of the filters to the input speech signal. Formally, for a given input $\mathbf{s}_t = \{s_{t-(kW-1)/2} \dots s_{t+(kW-1)/2}\}$, the output y_t of the first convolution layer is given by:

$$y_t[m] = \sum_{l=-(kW-1)/2}^{l=+(kW-1)/2} f_m[l] \cdot s_{t+l} \quad \forall m = 1,..,d_{out} \tag{4.4}$$

where f_m denotes the m^{th} filter in first convolution layer and $y_t[m]$ denotes the output of the filter at time frame t. Figure 4.3 presents the output of the filters of CNN-1H-mono WSJ given a segment of speech signal corresponding to phoneme $/I/$ as input. It can be seen that at each time frame only a few filters out of the 80 filters have high energy output. An informal analysis across different phones showed similar trends, except that the filters with high energy output were different for different phones. Together with the findings of the previous section, this suggests that the learned filters could be a *dictionary* that models the information in the frequency domain in parts for each phone. With that assumption, we extended the analysis where,

1. the magnitude spectrum or frequency response $\mathcal{S}_t$ of the input signal $\mathbf{s}_t$ based on the dictionary of learned filters is estimated as:

$$\mathcal{S}_t = |\sum_{m=1}^{M} y_t[m] \cdot \mathcal{F}_m|, \tag{4.5}$$

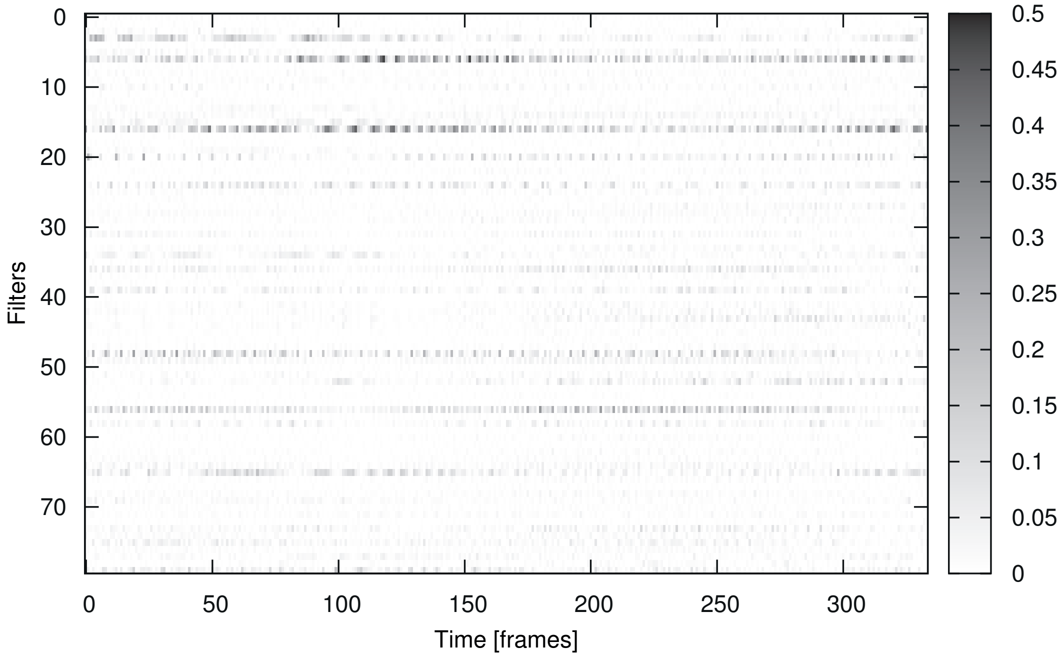


Figure 4.3 – Normalized energy output of each filter in the first convolution layer of CNN-1H-mono WSJ for an input speech segment corresponding to phoneme $/I/$.

where $y_t[m]$ is the output of filter m as in Equation (4.4) and $\mathcal{F}_m$ is the complex Fourier transform of filter f_m.

It is worth noting that if the filter-bank was to correspond to a bank of Fourier sine and cosine bases then $\mathcal{S}_t$ is nothing but the Fourier magnitude spectrum of the input signal $\mathbf{s}_t$. As $y_t[m]$ would be a projection on to the Fourier basis corresponding to discrete frequency m, and $\mathcal{F}_m$ would *ideally* be a Dirac delta distribution centered at the discrete frequency m.

2. gain-normalized magnitude spectrum $\mathcal{S}_t$ is averaged across different frames and speakers for each phone. The resulting average magnitude spectrums for the phones are then compared.

We performed the analysis on the validation data of WSJ, MP-DE and MP-FR using the filters in the first convolution layer of respective CNN-1H-mono. The log-magnitude spectrums are displayed for a few prominent vowels (notated in SAMPA format) for WSJ in Figure 4.4, for MP-DE in Figure 4.5 and for MP-FR in Figure 4.6. It can be observed that the average magnitude spectrum is capturing envelope of the sub-segmental speech. Furthermore, it is different for each vowel. The prominent spectral peaks could be related to the formants. However, a detailed formant analysis is practically infeasible for three main reasons:

(a) First, poor frequency resolution. The filters are operating on sub-segmental speech of about 1.8ms. This leads to poor frequency resolution. It can be also noticed from the

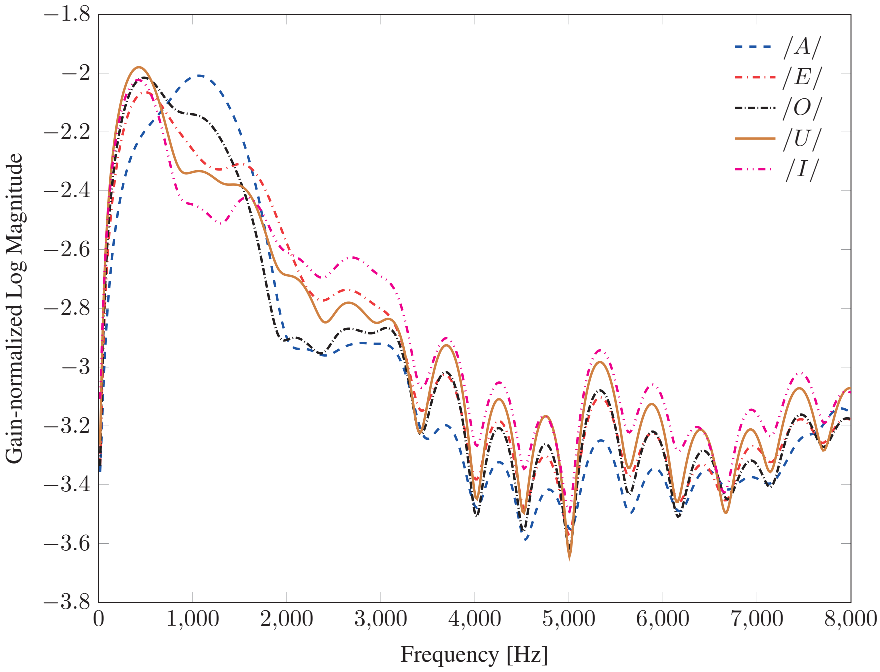


Figure 4.4 – Mean frequency response on the WSJ-mono corpus for phonemes $/E/$, $/A/$, $/O/$, $/I/$ and $/U/$.

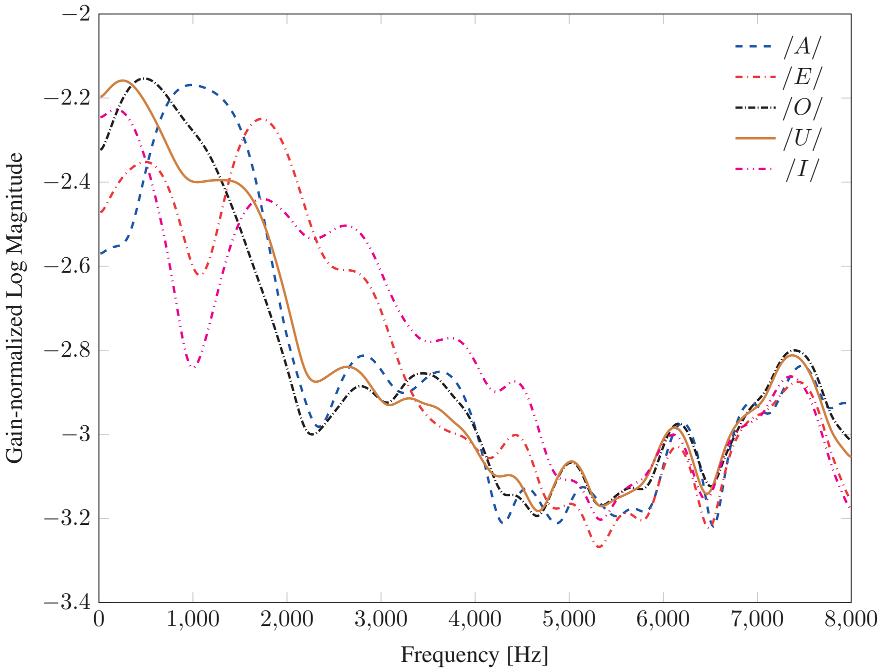


Figure 4.5 – Mean frequency response on the MP-DE corpus for phonemes $/E/$, $/A/$, $/O/$, $/I/$ and $/U/$.

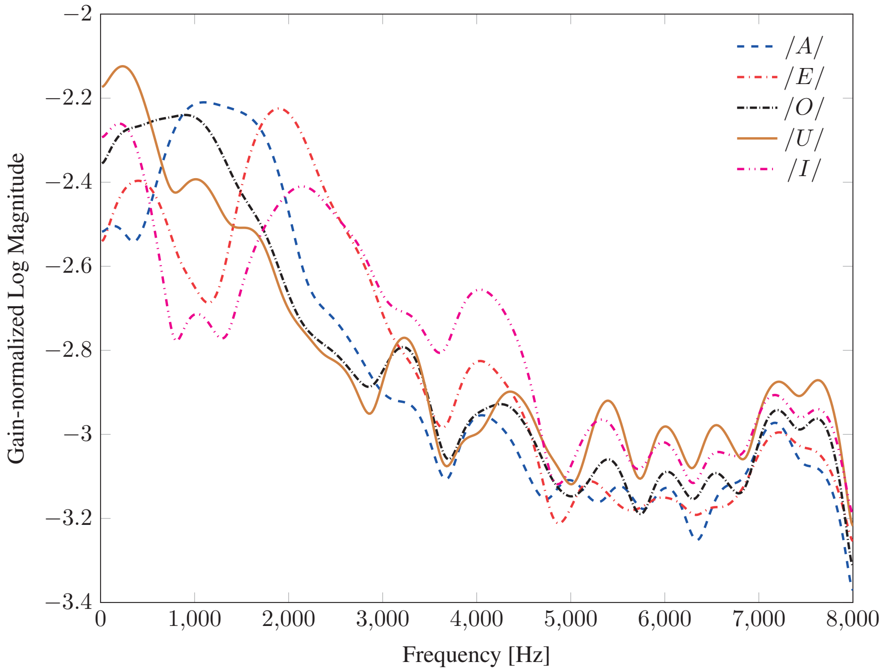


Figure 4.6 – Mean frequency response on the MP-FR corpus for phonemes /E/, /A/, /O/, /I/ and /U/.

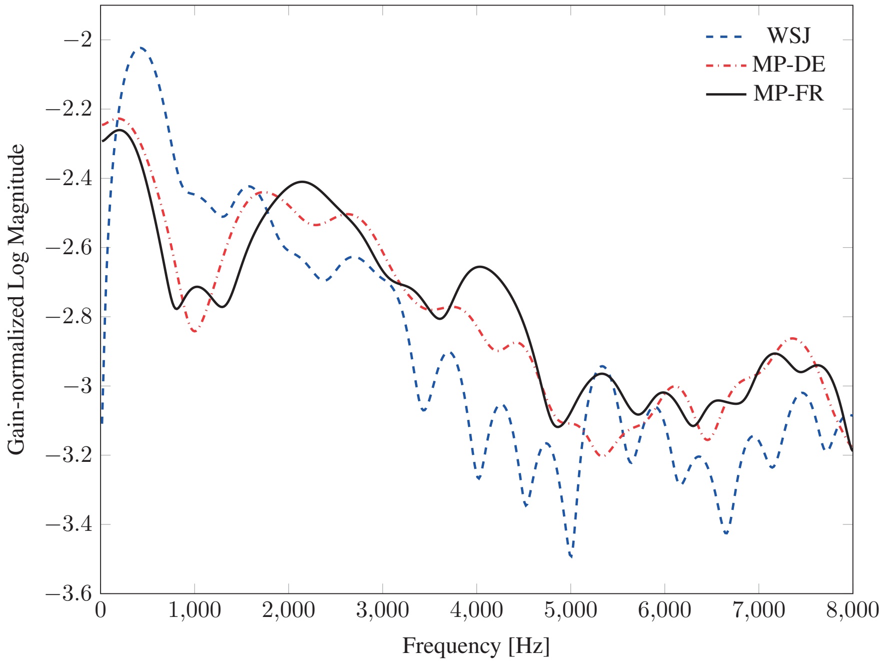


Figure 4.7 – Mean frequency response for English, German and French for phoneme /I/.

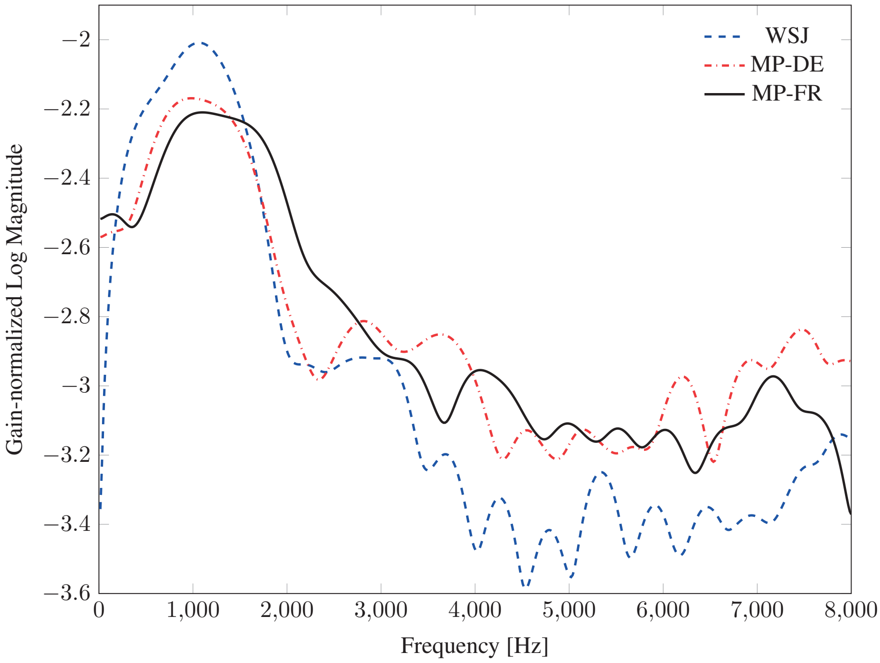


Figure 4.8 – Mean frequency response for English, German and French for phoneme /*A*/.

ripples in the magnitude spectrums (especially in the high frequency region);

(b) Second, the formant frequencies and their bandwidths for males and females are different. The frequency responses here are result of averaging over several male and female speakers in the respective validation data set; and

(c) Third, the analysis here has been carried on validation data, not on actual training data. So there can be spurious information present due to unseen condition or variation.

For instance, in the case of /A/, see Figure 4.8, we observe a prominent peak at around 1000 Hz, which could be seen as merger of first formant and second formant as a consequence of window effect and averaging over male and female speakers. Taking these aspects into account, we examined the frequency responses in the case of WSJ (Figure 4.4). We found that the prominent spectral peak locations tend to relate well to the first formant, second formant and third formant information provided for English vowels in [Deng and O'Shaughnessy, 2003, p. 233]. When comparing across the languages (Figure 4.7 and Figure 4.8) we observe a trend similar to the cumulative response of the filters (Figure 4.2). Specifically, the main peaks locations and spectral balance match well for MP-DE and MP-FR. However, in the case of WSJ the spectral peak locations tend to match but the spectral balance is different than MP-DE and MP-FR.

Given the understanding gained by the first convolution layer analysis and CNN architecture, it can be hypothesized that the second convolution layer model the modulation of the first layer filter outputs.

4.2 Intermediate Feature level Analysis

In this section, we focus on the analysis of intermediate feature representations that are being learned at the output of the feature learning stage. In that regard, Section 4.2.1 focuses on the discriminative aspects of the learned feature representations. Section 4.2.2 then focuses on the cross-domain and cross-lingual aspects.

4.2.1 Discriminative Features

In the recognition studies presented earlier in Section 3.3, it was observed that CNN-1H system with much fewer parameters outperforms ANN-3H system on all the tasks. Furthermore, we also observed that the capacity of the proposed CNN-based system lies more at the classifier stage. Given that the intermediate feature representations are learned in the process of training $P(i|\mathbf{s}_t^c)$ estimator, it can be presumed that these features are more discriminative compared to cepstral-based feature representations, and thus needs less parameters in the classifier stage. To fully ascertain that aspect we conducted an experiment to compare the cepstral features and the intermediate feature representations learned by the CNN. Specifically, we trained and tested three single layer perceptron based systems on WSJ task. One with the MFCCs with temporal context (39×9) as input and the others with intermediate features learned by CNN-1H and CNN-3H. In the case of CNN-3H, w_{in} was kept same as CNN-1H i.e. 210 ms. Table 4.1 presents the performances of the three systems. We can observe that the learned features lead to a better system than the cepstral features. Thus, indicating that the learned features are indeed more discriminative than the cepstral feature representation. Furthermore, it is interesting to note that the features learned by CNN-1H and CNN-3H yield similar systems. It suggests that the gain in ASR performance for WSJ task using CNN-3H is largely due to more hidden layers.

Table 4.1 – Single layer perceptron based system results on the Nov'92 testset of the WSJ task.

Features	Dimension	WER (in %)
MFCC	351	10.6
CNN-1H	540	7.9
CNN-3H	540	7.9

4.2.2 Cross-domain and Cross-lingual Studies

Conventional cepstral features, like MFCC, are known to be independent of the language or the domain, which is one of the main reason they become “standard” features. In the proposed system, the features are learned in a data-driven manner, thus they may have some level of dependencies on the data. In order to ascertain to what level the learned features are domain or language independent, we conducted cross-domain and cross-lingual experiments. More

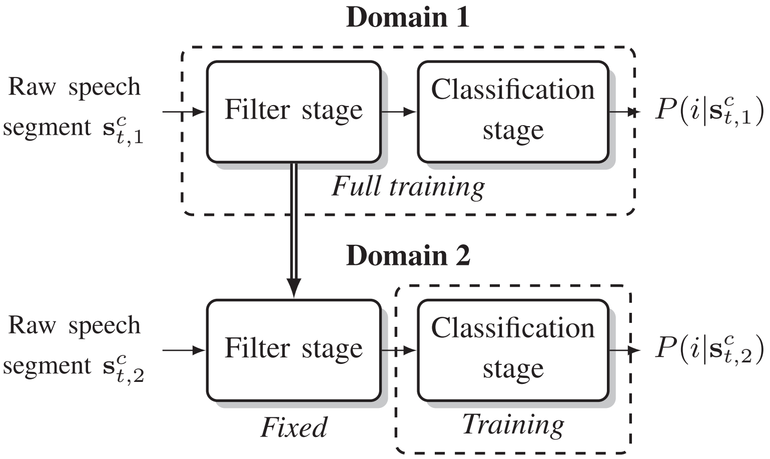

Domain 1
Raw speech segment $\mathbf{s}_{t,1}^{c}$
Filter stage
Classification stage
$P(i|\mathbf{s}_{t,1}^{c})$
Full training
Domain 2
Raw speech segment $\mathbf{s}_{t,2}^{c}$
Filter stage
Classification stage
$P(i|\mathbf{s}_{t,2}^{c})$
Fixed
Training

Figure 4.9 – Illustration of the cross-domain experiment. The filter stage is trained on domain 1, then used as feature extractor on domain 2.

precisely, as illustrated in Figure 4.9, in these experiments the filter stage is first trained on one domain or language. It is then used as feature extractor to train the classifier stage of another domain or language.

We used the TIMIT task and WSJ task for cross-domain experiments. We investigated

1. the use of feature stage of CNN-1H of WSJ task as feature extractor for TIMIT task. The classifier stage with single hidden layer was trained on TIMIT to classify 183 phone classes.
2. the use of feature stage of CNN-1H of TIMIT task as feature extractor for WSJ task. The classifier stage with single hidden layer was trained to classify 2776 clustered context-dependent units.

In both the studies, we set the number of hidden nodes to 1000, similar to the systems reported in Section 3.3. The results of the two studies are presented in Table 4.2. In the case of TIMIT task the results are presented in terms of PER, and in the case of WSJ task in terms of WER. In the TIMIT task, we can observe that, despite the feature stage being trained to classify clustered context dependent units on much larger corpus, the PER is inferior to the case where the feature stage is learned on TIMIT. In the case of WSJ task, we observe that with feature stage trained on TIMIT the WER is high.

In addition to the fact that TIMIT and WSJ are two different corpora, there are two other differences which could have had influence. First, WSJ is a much larger corpus than TIMIT in terms of data. Second, in TIMIT CNN-1H the feature stage is learned while classifying context-independent phones. Similarly in WSJ CNN-1H the feature stage is learned while classifying clustered context-dependent units. So, we conducted a study on WSJ task to understand the

Table 4.2 – Cross-domain results on English. The TIMIT results are in terms of PER. The WSJ task results are in terms of WER.

Classifier stage (Domain 2)	Feature stage (Domain 1)	Error Rate (in %)
TIMIT	Learned on TIMIT	22.8
	Learned on WSJ	23.3
WSJ	Learned on WSJ	6.7
	Learned on TIMIT	7.8

influence of the type of units at the output of the CNN on the feature stage learning, while negating the data effect. More precisely, we used the feature stage of WSJ CNN-1H-mono (presented earlier in Section 4.1.3) as feature extractor and trained the classifier stage to classify 2776 clustered context-dependent units. This system leads to a performance of 7.3% WER, which is inferior to 6.7% WER. This shows that indeed the type of units in the output of CNN has an influence on the feature learning stage. When compared to the case where the feature stage is learned on TIMIT, this result indicates that the majority of the performance gap can be attributed to the differences in the WSJ and TIMIT data sets. It is worth observing that TIMIT is a very small corpus compared to WSJ (3 hours vs 88 hours). However, the performance difference is not drastic, which suggests that the relevant features can be learned on relatively small amount of data.

We investigated the cross-lingual aspects on WSJ, MP-DE and MP-FR tasks. We conducted studies where the feature stage is learned on one language and the classifier stage is learned on the other language. For these studies, we used the feature stages of WSJ CNN-1H, MP-DE CNN-1H and MP-FR CNN-1H systems presented in Section 3.3. The classifier stage in all the studies consisted of a single hidden layer with 1000 nodes. The classes at the output of classifier stage remained same as before, i.e. 2776 cCD units for WSJ task, 1101 cCD units for MP-DE task and 1084 cCD units for MP-FR task. Table 4.3 presents the results of the study.

Before we analyze the results in detail, we can consider broader aspects. Specifically, in terms of family of languages, English and German belong to Germanic language family while French belongs to Romance language family. Given that, it can be expected that the feature stage learned on MP-DE to suit well for WSJ task when compared to feature stage learned on MP-FR and vice versa. In the case of WSJ task this trend is observed (12.1% vs. 12.8%). However, it is not observed in the case of MP-DE task (30.9% vs. 26.1%). In general we observe that feature stage learned on another language leads to inferior system. The performance gap is drastic when the feature stage is learned on WSJ and the classifier stage is learned on Medialparl (MP-DE or MP-FR) and vice versa. In addition to language differences, this can be attributed to the other differences in WSJ corpus and Medialparl corpus. More precisely, WSJ corpus contains read speech collected in controlled environment while Mediaparl contains spontaneous speech collected in real world conditions. This is also supported by the findings

Table 4.3 – Crosslingual studies result on English, German and French. The feature stage is learned on Domain 1 and the classifier stage is learned on Domain 2.

Classifier stage (Domain 2)	Feature stage (Domain 1)	WER (in %)
WSJ	Learned on WSJ	6.7
	Learned on MP-DE	12.1
	Learned on MP-FR	12.8
MP-DE	Learned on MP-DE	24.4
	Learned on MP-FR	26.1
	Learned on WSJ	30.9
MP-FR	Learned on MP-FR	25.9
	Learned on MP-DE	26.8
	Learned on WSJ	31.7

of the analysis presented in Section 4.1. Since MP-DE and MP-FR are similar kind of data except for the language, the drop in performance is small (24.4% to 26.1% in the case of MP-DE task and 25.9% to 26.8% in the case of MP-FR task). Languages typically have different phone sets and this difference gets further enhanced when modeling context-dependent phones. As we saw earlier in the cross-domain studies the choice of output units influences the feature stage. So, the small drop in performance in this case could be more attributed to the phonetic level differences between German language and French language.

4.3 Relation to Recent Literature

Recently, there are other works, inspired by ours, that have investigated modeling of raw speech signal directly using ANNs [Tüske et al., 2014, Golik et al., 2015, Sainath et al., 2015b]. In [Tüske et al., 2014], use of DNNs (or fully connected MLP) was investigated. It was found that such an acoustic model yields inferior system when compared to standard acoustic modeling. In a subsequent follow up work [Golik et al., 2015], it was found that addition of convolution layers at the input helps in improving the system performance and reducing the performance gap w.r.t standard acoustic modeling technique. In [Sainath et al., 2015b], an approach was proposed using convolutional long short-term memory deep neural network (CLDNN), where the input to CLDNN is raw speech signal. This approach was found to yield performance comparable to the case where the input to CLDNN is log filter bank energies. In comparison to these works, our work mainly differs at the feature stage or convolution layers. Specifically, in these works the short-term window size is set to about 16ms based on prior knowledge, while in our case it is a hyper-parameter and was determined to be around 2ms. Furthermore, in these works the filters learned at the first convolution layer were found to be similar to auditory filter-banks. In [Sainath et al., 2015b], these filters were close to Mel filter banks, while in [Golik et al., 2015] the filters were found close to well-known spectro-temporal filters, such as MRASTA filters [Hermansky and Fousek, 2005] and Gabor filters [Chang and Morgan, 2014]. In our case, the learned filters are a dictionary of matched filters that model formant-like information in the sub-segmental speech. As a whole, these works, similar to ours, show that the relevant features from the speech signal can be automatically learned along with the classifier to estimate $P(i|\mathbf{s}_t^c)$.

4.4 Summary

In this chapter, we presented an analysis of the features learned by the CNN-based system taking raw speech as input. We conducted the studies at two levels: on the filter level, i.e. the first convolution layer and on the intermediate representations level. Our studies showed that the first convolution acts as a filterbank and models "in-parts" the spectral envelope of short-term signal of 1.8 ms duration. The studies also showed that the learned features have some level of invariance across domains and languages. These learned features are also more discriminative than standard cepstral-based features. The following chapter further pursues this point.

5 Deep Features and Shallow Classifier

In pattern recognition, the trade-off between feature efficiency and classifier capacity[1] is well-known, and can be illustrated by two extreme cases. In the first case, if one assumes that the features represent the classes perfectly, the model can be as simple as possible. On the other hand, if the features are not robust, the model would need more capacity in the classifier. Typically, most systems operate at a middle point, where the feature are reasonably robust, so the classifier capacity is acceptable.

As discussed earlier in the thesis, the deep neural network approach consists of using NN-based classifiers with many hidden layers. The input of the DNNs is cepstral features or spectral-based features. It has been found that these systems improve with deep architecture, i.e. more hidden layers [Hinton et al., 2012]. However, the DNN approach of adding more layers has been questioned recently: as shown by Ba and Caruana [2014], shallow networks can be trained to perform similar to deep neural network. This raises the question: what is "deep"?

As presented in the earlier chapters, the CNN-based system using raw speech as input is able to learn relevant features in the filter stages. We also showed that the learned features are more discriminative than standard cepstral-based features. With respect to the features/classifier trade-off presented above, the CNN-based system seems to lean towards the efficient feature/simple classifier case. Motivated by these aspects, in this chapter we further study the capabilities of the CNN-based approach to learn efficient features using a simple classifier. More specifically, we investigate CNN-based architectures using *deep* features, i.e. many features learning layers and a *shallow* linear classifier. This approach has potential implications in controlling acoustic model capacity.

---

[1] In this chapter, we measure the capacity as the number of parameters of the classifier.

5.1 Architecture and Network Design

As done previously, the CNN-based system is composed of two stages: the features learning stage and the classifier stage. The filter stage is the same as the one described in Section 3.2.1, composed of a convolution layer, a max-pooling layer and a non-linearity. The classifier stage is a single layer perceptron (SLP), i.e. a linear classifier as opposed to MLP. Both stages are trained jointly using the approach presented in Section 3.2.2. Figure 5.1 illustrates the architecture of the proposed CNN-based acoustic model.

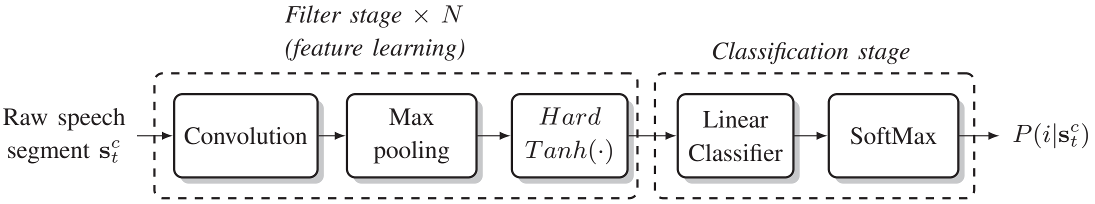


Figure 5.1 – Convolutional neural network based architecture using a linear classifier.

In this architecture, the capacity of the classifier cannot be tuned by a hyper-parameter, as it was the case in the previous chapters, because the classifier has no hidden layer. The classifier capacity is given by:

$$d_{out} \times N_{class} \tag{5.1}$$

where N_{class} denotes the number of output classes and d_{out} denotes the dimensionality of the intermediate representations, i.e. the output of the feature learning stage. Thus, the number of parameters of the classifier is entirely determined by d_{out}, given by:

$$d_{out} = N_{out} \times d \tag{5.2}$$

where N_{out} denotes the number of frames at the output of the last filter stage and d the dimension of these frames. d is a hyper-parameter, denoting the number of filters in the last convolution layer. The number of frames $N_{out,C}$ for an architecture using C filter stages is given by:

$$N_{out,1} = \frac{1}{kW_{mp,1}} \left(\frac{w_{in} - kW_1}{dW_1} + 1 \right) \tag{5.3}$$

$$N_{out,n} = \frac{1}{kW_{mp,n}} \left(\frac{N_{out,n-1} - kW_n}{dW_n} + 1 \right), \quad n = \{2, \ldots, C\} \tag{5.4}$$

where kW and dW are the hyper-parameters of the convolution layers, kW_{mp} and dW_{mp} are the hyper-parameters of the max-pooling layers and w_{in} is the input window, expressed in number of samples, as presented in Section 3.3.2

We can observe that the number of frames at the output of the feature learning stage N_{out} is actually decreasing when more filter stages are used, because of the non-overlapping max-pooling layers (Equation (5.4)). Thus, the capacity of the classifier decreases, see Equation (5.1), while the capacity of the filter stages increases. In this case, adding more features learning layers to the architecture has the effect of shifting the capacity of the whole system from the classifier stage to the feature learning stage.

5.2 Experimental Setup

We present two studies to demonstrate the potential of shifting the capacity to the features learning stage. The first study is a controlled study on TIMIT phoneme recognition task where the total number of parameters is fixed. The second study is a continuous speech recognition study on WSJ task, where the total number of parameters is variable.

Phoneme recognition study We first present a controlled study, where the depth of the feature learning stage is studied where the sum of parameters in the features stage and the classifier stage is a constant. As presented in the previous section, varying the depth of the feature learning stage has the effect of shifting the capacity from the classifier to the feature learning stage. In this study, we vary the depth of the features learning stage from one to four filter stages. We perform this study on phoneme recognition task on the TIMIT corpus, the details of the setup can be found in Section 3.3.1. The network hyper-parameters are carefully selected to fulfil the fixed capacity constraint using the validation data. The hyper-parameters selected are presented in Table 5.1. We first compare the architecture with a SLP-based hybrid HMM/ANN system, with 64k parameters. We then compare with a MLP-based system with one hidden layer of 500 units, which has 320k parameters. In this case, the number of parameters of the CNN-based system is fixed to 132k and 320k.

Continuous speech recognition study The objective of the second study is to evaluate the potential of shifting the capacity to the feature learning stage to *reduce* the capacity of the system on a large-scale task. In this study, we vary the depth of the feature learning stage from one to four filter stages. The hyper-parameters are tuned on a coarse grid search and presented in Table 5.1. The study is performed on WSJ continuous speech recognition, as presented in Section 3.3.1. We compare our system to HMM/ANN baselines using SLP and ANN-1H classifier, using MFCC features as input. We also compare the proposed architecture to the CNN-based system using MLP-based classifier, referred to as CNN-1H, presented in Chapter 3.

Table 5.1 – Network hyper-parameters.

Corpus	# conv. layer	# total params.	w_{in}	kW	dW	d_n	kW_{mp}
TIMIT	1	64k	310 ms	30	10	38	50
	2	64k	310 ms	30,5	10,1	40,34	7,7
	3	64k	310 ms	30,7,7	10,1,1	45,44,40	7,7,7
	4	64k	310 ms	30,9,9,9	10,1,1,1	52,40,40,40	3,3,3,3
TIMIT	1	132k	310 ms	30	10	80	50
	2	132k	310 ms	30,5	10,1	40,38	5,5
	3	132k	310 ms	30,7,7	10,1,1	90,70,60	4,4,4
	4	132k	310 ms	30,9,9,9	10,1,1,1	80,60,60,60	3,3,3,3
TIMIT	1	320k	310 ms	30	10	194	50
	2	320k	310 ms	30,5	10,1	100,85	5,5
	3	320k	310 ms	30,7,7	10,1,1	200,108,100	4,4,4
	4	320k	310 ms	30,7,7,7	10,1,1,1	150,120,100,90	3,3,3,3
WSJ	1	1.3M	310 ms	30	10	80	50
	2	1M	310 ms	30,7	10,1	80,40	7,7
	3	800k	310 ms	30,7,7	10,1,1	100,100,50	4,4,4
	4	590k	310 ms	30,7,7,7	10,1,1,1	80,60,60,60	3,3,3,3

5.3 Results

In this section, we first present the results of the study on TIMIT phoneme recognition task and then the WSJ task.

5.3.1 Phoneme Recognition Study

Table 5.2 presents the results of the CNN-based system compared to the SLP baseline, where the capacity is fixed to 64k parameters. The results are expressed in term of PER. We can see that the CNN-based system is able to yield similar performance to the baseline with only one convolution stage. Adding more filter stages shifts the capacity of the system from the classifier stage to the feature learning stage of the system and at the same time improves performance.

We compare the CNN-based system to the ANN-1H baseline with 320k parameters with a fixed number of parameters of 132k and 320k. The results are presented in Table 5.3. It can be observed that the CNN-based system using four filter stages outperforms the baseline with the same amount of parameters (320k). Moreover, the CNN also outperforms the baseline with less than half of the parameters (132k). When compared to the results presented in Chapter 3,

Table 5.2 – Results on the TIMIT core testset with 64k parameters.

Features	# conv. layers	# conv. param.	Classifier	# classifier param.	Total # params.	PER
MFCC	na	na	SLP	64k	64k	37.2 %
RAW	1	1k	SLP	63k	64k	37.7 %
RAW	2	8k	SLP	56k	64k	30.5 %
RAW	3	28k	SLP	36k	64k	29.3 %
RAW	4	50k	SLP	14k	64k	27.3 %

Table 5.3 – Results on the TIMIT core testset with 132k and 320k parameters.

Features	# conv. layers	# conv. param.	Classifier	# classifier param.	Total # params.	PER
MFCC	na	na	ANN-1H	320k	320k	25.6 %
RAW	1	2k	SLP	130k	132k	35.4 %
RAW	2	8k	SLP	124k	132k	30.9 %
RAW	3	76k	SLP	56k	132k	28.7 %
RAW	4	110k	SLP	22k	132k	25.4 %
RAW	1	6k	SLP	314k	320k	33.9 %
RAW	2	45k	SLP	275k	320k	28.3 %
RAW	3	233k	SLP	87k	320k	26.6 %
RAW	4	277k	SLP	43k	320k	25.2 %

5.3.2 Continuous Speech Recognition Study

The results for the study on continuous speech recognition task on the WSJ corpus are presented in Table 5.4 along with the SLP and MLP baselines results. The performance is expressed in terms of word error rate (WER). We observe a similar trend as in the TIMIT studies, i.e. the performance of the system improves with increase in filter stage capacity and reduction in the classifier stage capacity. More specifically, it can be observed that with only two convolution layers the proposed system is able to achieve performance comparable to the SLP baseline. With four convolution layers, the system is able to yield performance comparable to the ANN-1H baseline using MFCC as input and the CNN-1H system with six times fewer parameters.

5.4 Discussion and Summary

In this chapter, we investigated the trade-off between feature learning stage and classifier stage in the proposed CNN-based acoustic modeling approach. Our studies indicate that the capacity of the acoustic model can be effectively controlled or reduced by increasing the

Table 5.4 – Results on the Nov'92 testset of the WSJ corpus.

Features	# conv. layers	# conv. param.	Classifier	# classifier param.	Total # params.	WER
MFCC	na	na	ANN-1H	3M	3M	7.0 %
RAW	3	55k	CNN-1H	3M	3M	6.7 %
MFCC	na	na	SLP	1M	1M	10.6 %
RAW	1	5k	SLP	1.3M	1.3M	15.5 %
RAW	2	27k	SLP	1M	1M	10.5 %
RAW	3	108k	SLP	700k	800k	8.5 %
RAW	4	180k	SLP	410k	590k	6.9 %

depth of the feature learning stage using a simple linear classifier stage, while keeping the performance of ASR system intact. A question that arises is that: can the deep feature stage also be replaced by a shallow network? It seems not to be the case. As shown by Urban et al. [2016], unlike DNNs, it is not trivial to replicate the performance of deep CNNs by shallow CNNs.

In the literature, the issue of capacity has been addressed early since the emergence of neural networks. LeCun et al. [1989] proposed the optimal brain damage approach, where the network is iteratively pruned. More recently, model compression has been proposed by Bucilua et al. [2006]. In speech, recent works by Ba and Caruana [2014] and Hinton et al. [2015] show that similar performance are yielded by small networks trained using the knowledge acquired by large networks. Overall, in these works, the approach mainly consists of training a large network and then reducing its capacity. However in our case, the network is directly trained with small capacity and yields performance comparable to system with more capacity. This has potential implication in training or adapting systems on scarce data.

6 Towards Noise-Robust Raw Speech-based Systems

The previous chapters showed that using the raw speech signal as input leads to competitive ASR systems. It was also found that the features learned by such system tends to model the spectral envelop of the sub-segmental speech signal and yields some degree of invariance across languages. A natural question which arises from these previous findings is that: whether the proposed CNN-based approach using raw speech is robust to noise?

In this chapter, we propose a robust CNN-based architecture, referred to as Normalized Convolutional Neural Networks (NCNN). This architecture is based upon the CNN-based architecture presented in Section 3.2.1, where normalization layers are introduced at each filter stage, which normalize the intermediate representations learned by the network to have zero mean and unit variance. Such a normalization is analogous to feature mean and variance normalization, which has been shown to provide robustness to noise in conventional ASR systems [Furui, 1981].

In the remainder of this chapter, we provide a brief literature review. We then present the proposed architecture. The recognition studies are then presented, followed by an analysis.

6.1 Related Literature

Robustness to noise is an important aspect of ASR system. Noise can be defined as undesirable sounds or signals corrupting the speech signal. Noises can be grouped in two types: additive and convolutive noise. Additive noise is added to the signal and usually it originates from the environment. Convolutive noise represent the effect of the channel between the speaker and the receiver, which can be expressed as a convolution operation. In the literature, this problem has been approached in two different ways: Model-based approach and feature based approach. In this section, we provide a brief review.

Model-based approaches assume that the features are sensitive to noise, and aim to model this sensitivity by adapting the acoustic model. The most popular approach is multi-conditional training, where the training set is corrupted by a set of representative noise conditions [Furui, 1992]. Another model-based approach is the signal decomposition approach, which is based on modeling each decomposable component of the noisy signal by separate models [Varga and Moore, 1990]. The parallel model combination is a similar approach, where the separate models are combined for recognition [Gales and Young, 1996]. Multi-band processing has also been proposed, where each frequency band of the signal is modeled separately [Bourlard and Dupont, 1997]. Following a similar approach, multi-stream processing has been investigated for robust ASR [Bourlard et al., Hagen, 2001, Misra et al., 2006, Ikbal, 2004]. The missing data approach is based on selecting reliable regions of the signal to train the model [Cooke et al., 2001, Raj et al., 2001]. Vector Taylor series approach also has been proposed for noise compensation [Li et al., 2007].

The feature-based approach consists of enhancing the input features prior to recognition. An early method is the spectral subtraction [Boll, 1979], which is based on estimating the noise power spectrum. This approach has been extended to non-linear spectral subtraction [Lockwood and Boudy, 1992] and continuous spectral subtraction [Flores and Young, 1994]. It was later extended to unsupervised spectral subtraction [Lathoud et al., 2005]. The feature enhancement approach has also been investigated for cepstral-based features. The most popular approaches are the Cepstral Mean Normalization [Furui, 1981] and the Cepstral Variance Normalization [Viikki and Laurila, 1998]. More recently, SNR features [Garner, 2009] have also been proposed.

In ANN-based framework, the ANN has been used to extract robust features [Tamura and Waibel, 1988, Sharma et al., 2000, Vinyals and Ravuri, 2011]. Recently, DNN-based systems have been investigated for robust ASR. In [Seltzer et al., 2013], the DNN-based system is shown to outperform HMM/GMM systems in multi-condition training setup without any enhancement techniques. Feature enhancement techniques have also been investigated for DNNs, such as Vector Taylor Series [Li and Sim, 2013]. Recurrent Neural Networks have also been investigated for robust ASR [Weng et al., 2014], showing that such approach can outperform the DNN approach in the multi-conditional training setup.

Unlike the conventional approach, in the proposed CNN-based acoustic modeling approach the feature stages and the classifier are jointly learned. As observed earlier in Chapter 4, the first convolution layer models in part the formant-like information in the envelop of sub-segmental speech. These regions are typically high signal-to-noise ratio regions. Thus the CNN-based system can be expected to be less susceptible to noise. A possible way to further improve robustness would be to enhance intermediate representations. In the following section, we present an approach.

6.2 Normalized Convolutional Neural Networks (NCNN)

The Normalized CNN is based on the CNN presented in Section 3.2.1. It is composed of several filter stages, followed by a classification stage, as illustrated in Figure 6.1. The filter stage is composed of a convolution layer followed a max-pooling layer. The representations learned by

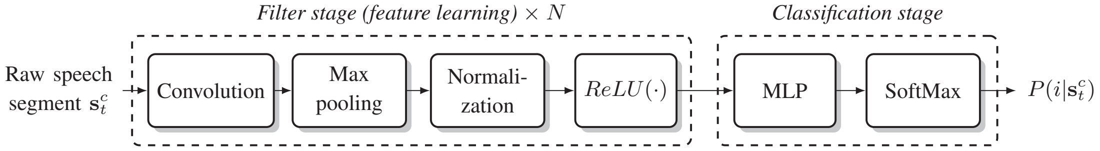


Figure 6.1 – Illustration of the normalized convolutional neural network architecture with N filter stages.

these layers are then given as input to a normalization layer. The normalized representations are then given as input to a non-linearity. The representations learned by these stages are then given as input to the classifier stage, composed of a standard MLP.

6.2.1 Normalization Layer

The normalization layers perform a temporal normalization over the input window w_{in} on each dimension of the outputs of the max-pooling layer, as illustrated in Figure 6.2. Formally, given the outputs of a max-pooling layer $O = \{\mathbf{o}_1 \cdots \mathbf{o}_N\}$ composed of N frames of dimension d_{out}, the normalization operation on one frame $\mathbf{o}_n$ is defined as:

$$Norm(\mathbf{o}_n) = \frac{\mathbf{o}_n[d] - \mu[d]}{\sigma[d]} \quad \forall d = \{1, \dots, d_{out}\} \tag{6.1}$$

where μ denotes the mean input vector, computed over all N frames,

$$\mu[d] = \frac{1}{N} \sum_{n=1}^{N} \mathbf{o}_n[d] \tag{6.2}$$

and the variance $\sigma^2[d]$ is computed using the unbiased variance estimation

$$\sigma^2[d] = \frac{1}{N-1} \sum_{n=1}^{N} (\mathbf{o}_n[d] - \mu[d])^2. \tag{6.3}$$

This normalization is applied on every output frame. It is worth mentioning that the number of output frames N can vary according to the position of the filter stage in the architecture, as presented earlier in Figure 3.6. It is worth mentioning that this layer was inspired by the batch normalization technique [Ioffe and Szegedy, 2015]. The key difference in our case is that the normalization is performed over time, not over a batch of examples.

6.2.2 Rectifier Linear Unit

Irrespective of whether normalization layers are employed or not in the CNN, we use the Rectified Linear Unit ($ReLU$) [Nair and Hinton, 2010] as non-linearity instead of the $HardTanh$, as it has been shown to bring robustness to DNN-based systems [Sivadas et al., 2015]. The

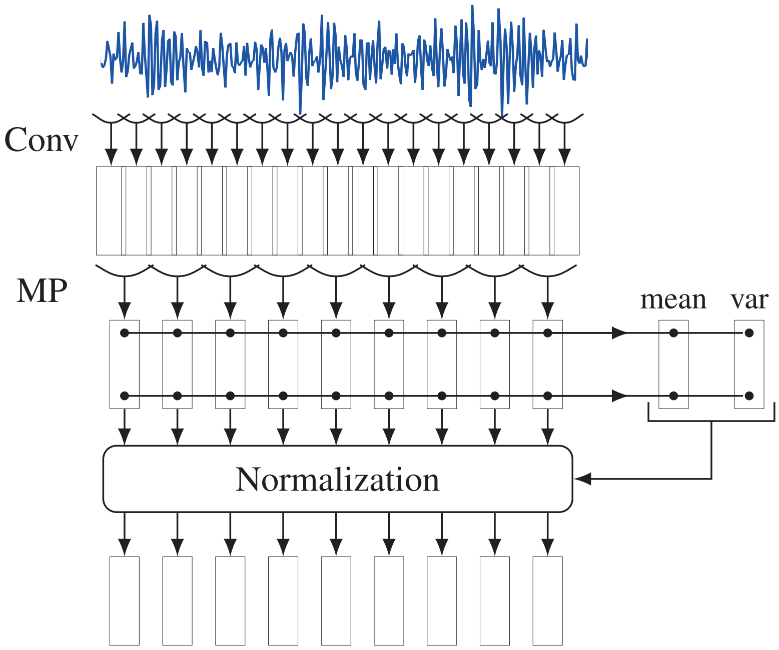


Figure 6.2 – Illustration of the first filter stage of the normalized convolutional neural network.

Rectifier Linear Unit is defined as:

$$ReLU(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases} \tag{6.4}$$

6.3 Connected Word Recognition Study

In this section, we present our studies on Aurora2 benchmark corpus.

6.3.1 Database

The Aurora2 corpus [Hirsch and Pearce, 2000] is a connected digit corpus which contains 8,440 sentences of clean and multi-condition training data, representing around 4 hours of speech, and 70,070 sentences of clean and noisy test data, sampled at 8 kHz. We report the results on test A and test B, composed of 10 different noises at 7 different noise levels (clean, 20dB, 15dB, 10dB, 5dB, 0dB, -5dB), totaling 70 different test scenarios, each containing 1,001 sentences. The alignment is obtained using the HTK-based HMM/GMM system provided along with the database. It consists of whole word HMM models with 16 states per word to model the digits. The states are connected in a simple left-to-right fashion. The number of states is 179.

6.3.2 Baselines

We compare our approach with the HMM/GMM baseline provided with the corpus [Hirsch and Pearce, 2000], which uses 16 Gaussian per state, and 179 states. We also train a HMM/ANN system, where the ANN has one hidden layer of 1000 units. As per the protocol, 39 dimension MFCCs input features are used, computed using HTK. The cepstral mean and variance normalization techniques are also used, applied on each utterance separately. These normalized features are referred to as MFCC-CMVN. We also consider the case where the speech signal is enhanced using Advanced Front End (AFE) tool [Hirsch, 2002b, Hirsch and Pearce, 2006] and MFCC are extracted. AFE is an ETSI standard describing the front-end of a distributed speech recognition system. It consists of a waveform noise reduction stage followed by a MFCC extractor.

6.3.3 CNN-based Systems

In these studies, we compare two architectures: the NCNN architecture, presented earlier in Section 6.2 and the CNN architecture, described in Section 3.2.1, except that the non-linearity is the $ReLU$ instead of the $HardTanh$. The network hyper-parameters defining the CNN architecture were based on the studies performed in Chapter 3. They are presented in Table 6.1.

Table 6.1 – Architecture of CNN-based system for the Aurora2 studies. w_{in} is expressed in terms of milliseconds. The hyper-parameters kW, dW, d_{out} and kW_{mp} for each convolution layer is comma separated. HU denotes the number of hidden units.

HL	w_{in}	kW	dW	d_{out}	kW_{mp}	HU
1	310	30,7,7	10,1,1	80,60,60	3,3,3	1000

The input of the CNN is a window of the speech signal normalized such that it has zero mean and unit variance, as described earlier in the studies presented in Chapter 3. We also investigate the case where the speech signal is enhanced before being fed to the CNN using AFE tool. This was done by taking the output of the two stage Weiner filter with the AFE tool and performing overlap add [Allen and Rabiner, 1977] followed by one bit dithering.

6.3.4 Results

We report the results in term of word recognition rate (WRR), on the clean test set, the test set A and test set B (as defined in Section 6.3.1). As per the protocol, the average values for all noise conditions between 0 and 20 dB are reported for test set A and test set B.

Original Waveforms

Table 6.3 presents the results on the clean and multi-conditional training setup. On the clean training, the CNN system yields better performance than the baseline system using MFCC features (without CMVN). The proposed NCNN approach outperforms the original CNN architecture by 18 % in absolute. It also outperforms the baselines systems using MFCC features. When compared to baseline using the CMVN technique, the NCNN approach clearly outperforms the HMM/GMM system, and yields similar performance to the HMM/ANN baseline. On the multi-conditional training setup, the NCNN and the CNN systems outperform the baselines, with and without using the CMVN techniques. Also, it can be observed that the NCNN system yields similar performance to the CNN-based system, which indicates that the normalization layers might not be necessary in the multi-conditional training setup. When compared to the literature, the proposed CNN-based system yields similar performance, as presented in Table 6.2.

Table 6.2 – Comparison with the literature on Aurora2 multi-conditional training setup, expressed in Word recognition rate (WRR). DAE stands for denoising auto-encoder and DVAT stands for discriminative adaptive training using vector Taylor series.

System	Clean	Test A	Test B
NCNN	98.95	94.23	92.24
Recurrent DAE [Maas et al., 2012]	99.06	89.78	82.52
DVAT HMM/GMM [Ragni and Gales, 2012]	-	95.4	95.1

Enhanced Waveforms

Table 6.4 presents the results using the enhanced waveforms in clean and multi-conditional training setup. In clean training setup, one can see that using the AFE enhancement technique on MFCC features improves the performance of the baseline systems. It can also be observed that RAW AFE inputs improves the performance of the CNN system by 11% on test set A and by 7.8% on test set B. However, using enhanced waveforms on the NCNN system actually decreases the performance by 2% on test sets A and B. This could be explained by the presence of artefact in the denoised waveforms of the training set. To confirm that aspect, we ran an experiment where the NCNN system is trained using original waveforms, and the test set are denoised. Using this setup, we see an improvement in performance for both tests: the NCNN system yields 87.1% WRR on test set A and 85.9% WRR on test set B. We also see an improvement compared to the case where original waveforms are used (see Table 6.3). However, either way the NCNN and the CNN system yield lower performance than the baseline.

In the multi-conditional training setup, one can see that the CNN system and the NCNN system outperforms all the baselines. When compared to using the original waveforms, both system yields similar performance, indicating that the waveform enhancement might not be necessary in this case. To understand the effect of AFE enhancement technique in the multi-conditional training setup, we ran the same experiment, where only the test sets are enhanced using AFE tool. Surprisingly, the performance is worse: the NCNN system yields 88.5% WRR on test set A and 86.0% WRR on test set B, which represents a drop in performance of about 7%. The same trend can be observed for the CNN system, which yields 83.6% WRR on test set A and 80.5% WRR on test set B, representing a performance drop of about 12%. A possible reasoning for this trend could be that the AFE tool was developed considering MFCC extraction with subsequent post-processing and its transmission for distributed speech recognition. This could be partly observed when comparing baseline systems with and without CMVN. Specifically, without any speech enhancement, the baseline systems improve with CMVN. However, with speech enhancement the performance of the baseline systems actually drop with CMVN.

Table 6.3 – Word recognition rate (WRR) on the Aurora2 test sets. HMM/GMM baseline performance using MFCC are reported in [Hirsch and Pearce, 2000] and the HMM/GMM baseline performance using MFCC-CMVN are reported in [Garner, 2009].

		Clean training			Multi-cond. training		
Features	System	Clean	Test A	Test B	Clean	Test A	Test B
MFCC	HMM/GMM	99.02	61.34	55.74	98.52	87.81	86.27
	HMM/ANN	99.13	60.96	64.63	98.47	92.14	82.37
MFCC-CMVN	HMM/GMM	99.13	77.98	78.78	97.97	90.94	90.75
	HMM/ANN	99.50	85.79	85.20	98.69	93.36	90.68
RAW	CNN	99.44	69.10	66.37	99.04	94.20	92.22
	NCNN	99.36	86.64	84.92	98.95	94.23	92.24

Table 6.4 – Word recognition rate (WRR) on the Aurora2 test sets, using enhanced waveforms. HMM/GMM baseline performance using MFCC are reported in [Hirsch and Pearce, 2006].

		Clean training			Multi-cond. training		
Features	System	Clean	Test A	Test B	Clean	Test A	Test B
MFCC	HMM/GMM	99.22	87.74	87.09	99.21	92.29	91.77
	HMM/ANN	99.37	78.96	76.32	99.30	94.11	92.10
MFCC-CMVN	HMM/GMM	99.15	88.73	89.23	98.81	90.83	89.64
	HMM/ANN	99.46	86.77	85.91	99.05	93.69	91.84
RAW AFE	CNN	99.37	80.26	74.21	99.12	95.08	93.31
	NCNN	99.35	84.64	82.96	98.91	94.32	92.99

6.4 Continuous Speech Recognition

In this section, we present the continuous speech recognition study on the Aurora4 corpus. This corpus is a subset of the Wall street journal corpus used in Chapter 3, corrupted with additive and convolutive noises.

6.4.1 Database

The Aurora4 corpus has been created from the standard Wall Street Journal (WSJ0) corpus, corrupted with six additive noises. The training set consists of 7180 utterances, representing 15 hours of speech. Two training conditions are provided: clean and multi-conditional training. The validation set is composed of 330 utterances. The data is sampled at 16 kHz. The test set is composed of 330 utterances. 14 conditions are provided, consisting of two different channels conditions. The test set is split into 4 subsets. Test A consists of the clean condition test (condition 1). Test B consists of the noisy utterances using a matched channel (conditions 2-7). Test C consists of the clean utterances using a mismatched channel (condition 8). Finally, Test D consists of noisy utterances using a mismatched channel (conditions 9-14). More details can be found in [Hirsch, 2002a]. The dictionary is based on the CMU phoneme set, 40 context-independent phonemes. We obtained 3000 clustered context-dependent (cCD) units, i.e. tied-states, by training a context-dependent HMM/GMM system with decision tree based state tying. We used the bigram language model provided with the corpus. The test vocabulary contains 5000 words.

6.4.2 Baselines

We compare our approach with the HMM/GMM system. We also train a HMM/ANN system, where the ANN has one hidden layer of 1000 units. Both systems use 39 dimension MFCC features, computed using HTK. Again, we investigate the case where cepstral mean and variance normalization of the features is performed at utterance level. These normalized features are referred to as MFCC-CMVN.

6.4.3 CNN-based Systems

As in the previous study, we compare two architectures: the NCNN architecture and the CNN architecture where the non-linearity is the $ReLU$ non-linearity instead of the $HardTanh$. The hyper-parameters of the features stage are based on the hyper-parameters found for the WSJ study presented in Section 3.3.2 and are presented in Table 6.5.

Table 6.5 – Architecture of CNN-based system for the Aurora4 studies.

HL	w_{in}	kW	dW	d_{out}	kW_{mp}	HU
1	310	30,7,7	10,1,1	80,60,60	3,3,3	1000

6.4.4 Results

The results on the Aurora4 corpus on the clean condition training setup are presented in Table 6.6, expressed in terms of word recognition rate. On test A (clean condition), one can see that the CNN system and the NCNN system yield similar performance to the baseline systems, with the CNN system sightly outperforming the NCNN system. On test set B (additive noise), one can see that the CNN barely reaches the performance of the HMM/GMM baseline using MFCC as input. The NCNN system outperforms all the baseline systems and the CNN system. On test set C (channel noise), the CNN system yields again similar performance to the GMM baseline with MFCC. The NCNN performance is on par with the baseline systems using MFCC CMVN. This suggests that the normalization layers are not very efficient for handling convolutional noise or channel effect. On test set D (additive and convolutional noise), the CNN system is outperformed by baseline systems using CMVN techniques and the NCNN outperforms all the baseline systems.

Table 6.7 presents the results using the multi-condition training setup. It can be observed that the baseline systems performance improves when using the CMVN techniques. On the four test sets, the NCNN system and the CNN system outperform all baselines. Interestingly, in this case the CNN system outperforms the NCNN system. This confirms the findings on the Aurora2 corpus that the normalization layers are not needed in the multi-conditional training setup.

Table 6.8 presents a comparison with the recent literature, where the proposed CNN system is compared on the multi-conditional training setup with a DNN-based system with 5 hidden layers (DNN-5H) and a CNN-based system with 4 hidden layers (CNN-4H), both using Mel-Filterbank energies as input. It can be observed that the CNN-based approach yields similar or better performance than these systems.

6.5 Analysis

In order to better understand the proposed NCNN architecture, we present in this section three analyses. We first analyze the role of the *ReLU* and the normalization layers on the robustness of the system. We then analyze the filters learned by the first convolution layer in clean and multi-conditional training setup. Finally, we study the effect of the AFE enhancement technique with respect to signal-to-noise ratio conditions.

Table 6.6 – Word recognition rate of the Aurora4 test sets on the clean training setup.

Features	System	Test A	Test B	Test C	Test D	Ave.
MFCC	HMM/GMM	90.73	41.72	51.65	25.65	52.43
	HMM/ANN	90.19	35.91	44.42	25.91	49.10
MFCC CMVN	HMM/GMM	93.14	56.25	61.29	35.76	61.61
	HMM/ANN	91.61	57.63	67.78	40.90	64.48
RAW	CNN	93.61	40.23	53.71	25.24	53.19
RAW	NCNN	92.02	77.57	66.88	51.47	71.98

Table 6.7 – Word recognition rate of the Aurora4 test sets on the multi-conditional training setup.

Features	System	Test A	Test B	Test C	Test D	Ave.
MFCC	HMM/GMM	84.81	72.91	52.29	55.55	66.39
	HMM/ANN	86.29	73.59	75.51	58.56	73.48
MFCC CMVN	HMM/GMM	89.28	79.80	78.11	63.10	77.57
	HMM/ANN	89.39	78.34	79.88	62.71	77.58
RAW	CNN	92.10	88.06	84.49	74.28	84.73
RAW	NCNN	91.74	87.50	83.43	73.22	83.97

Table 6.8 – Comparison with literature on the multi-conditional training setup.

Feature	System	Test A	Test B	Test C	Test D	Ave.
Mel-filterbank	DNN-5H [Mitra et al., 2014]	89.7	84.1	84.8	74.8	83.35
Mel-filterbank	CNN-4H [Mitra et al., 2014]	90.0	85.6	86.6	78.1	85.07
Raw	Proposed CNN	92.1	88.1	84.5	74.3	84.73

6.5.1 Architectures Analysis

When compared to the CNN-based system presented in Chapter 3, the NCNN architecture has two differences: the non-linearity and the normalization layers. In this section, we analyze their role on the robustness of the system on Aurora2.

Normalization We first evaluate the effect of the normalization layers by comparing the NCNN architecture, i.e. where the normalization is applied at each filter stage, with an architecture where the normalization is only applied in the last convolution layer. The results are presented in Table 6.9. One can see that applying a normalization at each layer clearly improves the performance in clean training setup. The performance in multi-conditional training setup is similar for both cases, supporting the argument that the normalization is not necessary in this setup.

Non-linearity We then evaluated the effect of the non-linearity layer. We compared the $ReLU$ layer to the $HardTanh$ layer, as defined in Section 6.2 of the present chapter and in Chapter 3 respectively. We trained a CNN system with $HardTanh$ and compared it to the CNN system with the $ReLU$ non-linearity. The results are presented in Table 6.10. The $ReLU$ non-linearity clearly leads to better system the $HardTanh$ non-linearity on clean and multi-conditional training setup.

Table 6.9 – Word accuracy on the Aurora2 corpus for different normalization strategies.

	Clean training			Multi-cond. training		
Normalization	Clean	Test A	Test B	Clean	Test A	Test B
At every filter stage	99.36	86.64	84.92	98.95	94.23	92.24
At the last filter stage	99.42	77.16	76.77	98.85	94.37	92.41

Table 6.10 – Word accuracy on the Aurora2 corpus for different non-linearities.

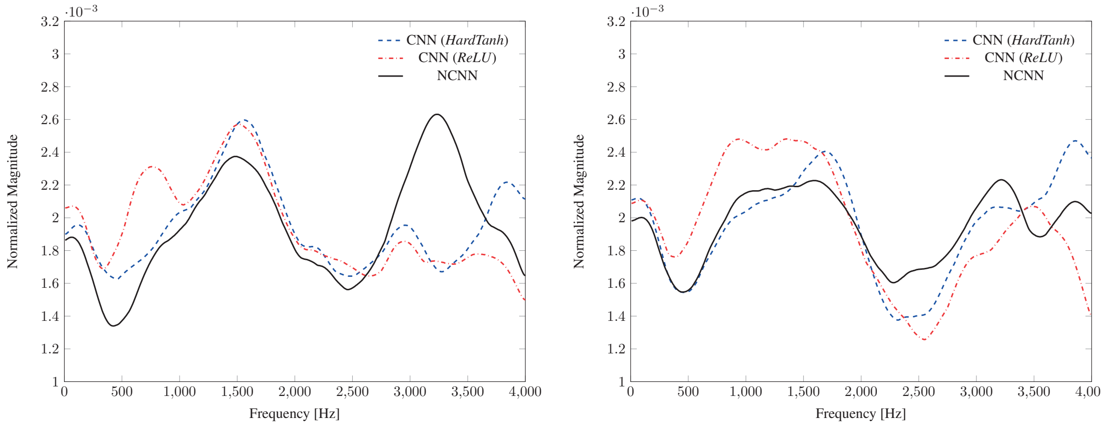
	Clean training			Multi-cond. training		
Non-linearity type	Clean	Test A	Test B	Clean	Test A	Test B
$ReLU$	99.44	69.10	66.37	99.04	94.20	92.22
$HardTanh$	99.34	67.68	64.20	98.66	93.45	90.76

6.5.2 First Convolution Layer Analysis

In order to gain further insights on the effects of the normalization, we computed the cumulative responses of the first convolution layer, as earlier done in 4.1.2. Specifically, we compared the NCNN architecture response with the CNN architecture responses using $HardTanh$ and $ReLU$ as non-linearity.

The cumulative frequency responses on Aurora2 are presented in Figure 6.3 for the CNN architecture using $HardTanh$ as non-linearity, using $ReLU$ as non-linearity and for the NCNN architecture. On the clean training setup, presented in Figure 6.3(a), one can see that at low frequencies, the three responses are close, they all have emphasis around 1.5 kHz. At high frequencies however, the NCNN response is different mainly around 3.3 kHz. This region is not emphasized by the CNN systems, and could explain the performance difference in the clean training setup.

The responses of the systems trained using the multi-conditional setup, presented in Fig-



(a) (b)

Figure 6.3 – Cumulative frequency responses on the Aurora2 corpus on (a) clean training, (b) multi-conditional training.

ure 6.3(b), show that the spectral balance is similar between the three systems. There is slight differences at high frequencies between the NCNN system and the CNN systems. When compared to the clean training setup, the spectral balance is different between the two training setups, as in multi-conditional training, the responses are more balanced across the whole spectrum. This can be explained by the fact that effect of noise tend to spread across all frequencies. One can also see that the emphasis around 1.5 kHz is flat on multi-conditional training setup. We also see that on both clean condition and multi-condition, the NCNN system lays emphasis around 3.0 - 3.5 kHz. Note that in this study, the CNN-based system classifies word states so relating these responses to phonemes is difficult.

The frequency responses on Aurora4 are presented in Figure 6.4. Before going into the details, it is worth noting that the response of the CNN system using the $HardTanh$ non-linearity on clean training matches the response on the WSJ corpus, presented in Figure 4.2. Using the clean training setup, presented in Figure 6.4(a), we can see that the responses of the three systems are close at low frequency and mismatch at high frequency. This is consistent with the Aurora2 findings. Using the multi-conditional training setup, a similar trend to the responses using clean conditions training can be observed. In fact, the frequency emphasis are consistent across training setups. There is however a difference in the spectral balance.

6.5.3 Waveform Enhancement Study

As presented in Section 6.3.4, use of the AFE waveforms enhancement technique with the NCNN systems leads to a drop in performance in the clean training case and do not improve performance in multi-conditional training setup. In order to understand the effect of the AFE technique, we analyzed the performance of the NCNN system with respect to the SNR level, using original and enhanced waveforms during training. We also analyzed the performance

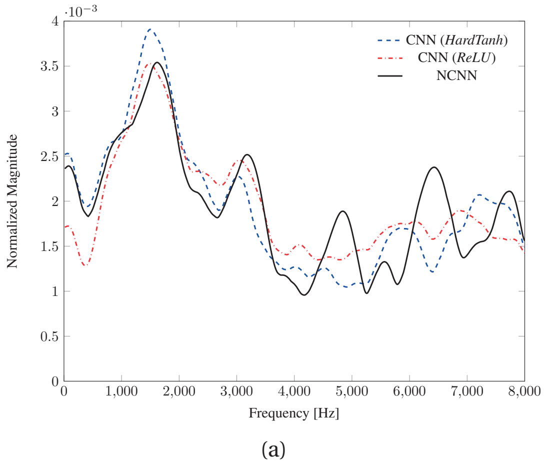


Figure 6.4 – Cumulative frequency responses on the Aurora4 corpus on (a) clean training and (b) multi-conditional training.

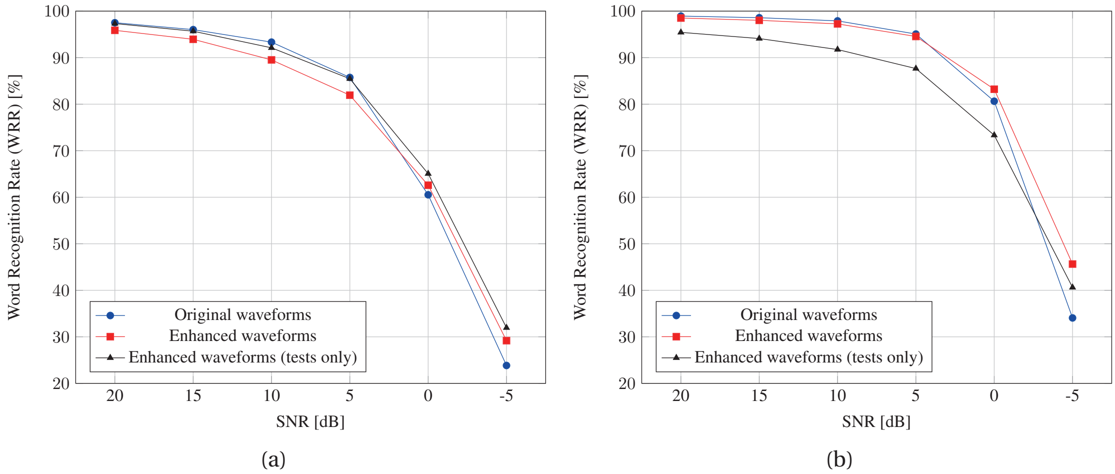


Figure 6.5 – Comparison of recognition performance on the test set A of Aurora2 using original and enhanced waveforms, on (a) the clean training setup and (b) the multi-conditional training setup.

when the system is trained with original waveforms and tested with enhanced waveforms. In the clean training setup, presented in Figure 6.5(a), using enhanced waveforms improves the performance only at very low SNR (0 and -5 dB) levels and decreases the performance at others SNR levels. The same trend can be observed when using enhanced waveforms only during testing, although there is a slight improvement at high SNR level. On the multi-conditional training setup, presented in Figure 6.5(b), using enhanced waveforms also only improves the performance at low SNR level. Using the enhancement waveforms leads to a performance drop at almost all SNR levels. This is consistent with our previous findings on Aurora2 in Section 6.3.4. Overall, these studies show that the benefit of using enhancement technique with data-driven feature learning approaches is not clear, i.e. open for further research.

6.6 Summary

In this chapter, we investigated the robustness of the CNN-based system to noise. To this aim, we proposed a novel approach based on intermediate representation normalization. Our studies showed that the proposed approach outperforms the baseline systems using feature level normalization. Furthermore, the studies also showed that the normalization layer is not needed when the CNN-based system is trained on multi-conditional dataset. Finally, we also investigated waveform enhancement using AFE tool on Aurora2 and we did not observe any benefit for the CNN-based system.

7 End-to-end Phoneme Sequence Recognition

In chapters 3 to 6, we investigated an end-to-end training approach applied to acoustic modeling in the hybrid HMM/ANN framework. We showed that such approach yields competitive performance on speech recognition tasks. In this framework, the phoneme sequence prediction is performed in two steps: first, the CNN-based acoustic model *locally* estimates the acoustic likelihood for each segment of the input speech utterance. In a second step, the sequence is decoded by the HMM, often using a language model. This approach is thus locally discriminative but globally generative. The training and the recognition are performed by maximizing $P(L,S)$, where S denotes the speech utterance and L its corresponding label sequence. Following the end-to-end approach, can we go one step further and train jointly the features, the classifier and the sequence decoding step?

In this chapter, we investigate an acoustic sequence to phoneme sequence conversion model, which takes a raw speech utterance as input and outputs a sequence of phoneme. This model consists of a local CNN-based classifier followed by a Conditional Random Fields (CRF). The system is trained based on the Graph Transformer Network [Bottou et al., 1997] approach, where the cost function discriminates the ground-truth sequence from all possible sequences. We investigate the approach in a systematic manner through three studies,

1. Separate training: In this system, the local classifier (CNN) and global sequence modeling (CRF) are trained separately, like in the hybrid approach.

2. Joint training: The system is trained in an end-to-end manner, where the CRF back-propagates the error gradient to the CNN-based classifier.

3. Weakly-supervised training: In separate training and joint training, we assume that the segmentation is available. In this system, we go one step further and investigate a training setup where only the phoneme transcription is available, not the segmentation. We extend the joint training approach to *simultaneously* infer the phoneme sequence segmentation and prediction.

7.1 End-to-end Sequence Recognition

The proposed system is composed of two stages: the sequence acoustic model based on Convolutional Neural Network, and the decoder, based on Conditional Random Fields. As illustrated in Figure 7.1, both stages are trained jointly through back-propagation.

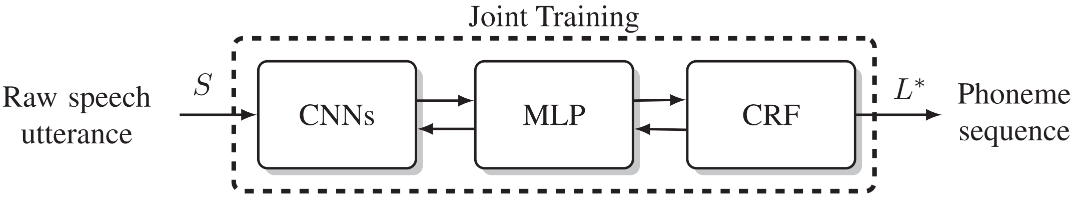


Figure 7.1 – Illustration of proposed system.

Acoustic Modeling

The acoustic modeling stage models a whole speech utterance. It is composed of the CNN-based architecture presented in Chapter 3. This stage is given a raw speech utterance S as input and outputs a score $f_t^i(S,\theta_f)$ for each class $i \in \{1,\dots,I\}$ at each frame t, where θ_f denotes the parameters of the networks.

Sequence Decoding

For the sequence decoding, we consider a simple CRF, where we define a graph with nodes for each frame in the input sequence, and for each label. Transition scores, denoted as a matrix A, are assigned to the edges between phonemes, and network prediction scores $f(\cdot)$ are assigned to the nodes. This CRF allows to discriminatively train a transition model over the network output scores. Given an input sequence S and a label path $L = \{l_1 \dots l_T\}$, $l_t \in \{1,\dots,I\}$, of length T on the graph, a score for the path can be defined:

$$c(S,L,\Theta) = \sum_{t=1}^{T} \left(f_t^{l_t}(S,\theta_f) + A_{l_t,l_{t-1}} \right) \tag{7.1}$$

where $\Theta = \{\theta_f,\ \theta_A\}$ denotes the parameters, with θ_f the CNN parameters and θ_A the CRF parameters, i.e. the matrix A. An illustration is provided in Figure 7.2. At inference time, the best label path can be found by maximizing (7.1). The Viterbi algorithm is used to find

$$L^* = \underset{L}{\operatorname{argmax}}(c(S,L,\Theta))\,. \tag{7.2}$$

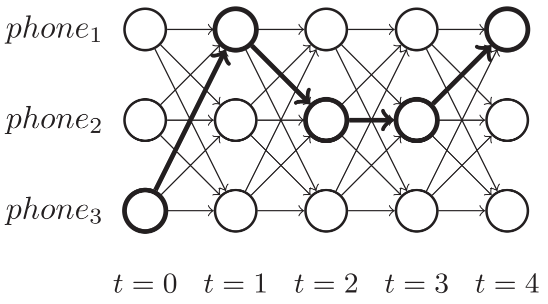


Figure 7.2 – Illustration of the CRF graph for 3 classes.

7.1.1 Supervised Training

In supervised training, we assume that the phoneme segmentation L is available during training. The system parameters Θ are learned by maximizing the log-likelihood $\mathcal{L}$, given by:

$$\mathcal{L}(\Theta) = \sum_{n=1}^{N} \log(P(L_n|S_n, \Theta)) \tag{7.3}$$

for each input speech sequence S and label sequence L over the whole training set. In a standard CRF setup, scores $c(S, L, \Theta)$ are interpreted as a conditional probability $P(L|S, \Theta)$ by taking them to the exponential (such that there are positive) and normalizing them over all possible label paths U in the fully connected lattice $\mathcal{U}_T$ of length T:

$$\log(P(L|S, \Theta)) = c(S, L, \Theta) - \underset{U \in \mathcal{U}_T}{\text{logadd}}\, c(S, U, \Theta), \tag{7.4}$$

where the logadd operation is defined in Equation (2.22).

Minimizing the negative likelihood $\mathcal{L}$ is performed using the stochastic gradient descent algorithm, where the parameters are updated by making a gradient step:

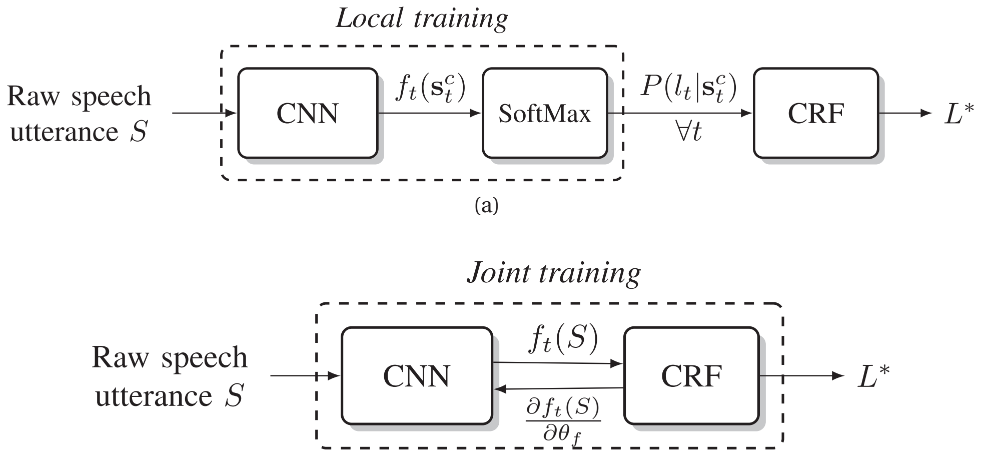
$$\Theta \longleftarrow \Theta + \alpha \frac{\partial \mathcal{L}(\Theta)}{\partial \Theta} \tag{7.5}$$

where α is the learning rate.

Two training strategies are considered: joint training and separate training. They are illustrated in Figure 7.3.

Joint Training

The networks and the CRF are trained jointly. In this case, The likelihood $\mathcal{L}$ is optimized with respect to the CRF parameters θ_A and to the network parameters θ_f, as presented above. The


Local training
Raw speech utterance S
CNN
$f_t(\mathbf{s}_t^c)$
SoftMax
$P(l_t|\mathbf{s}_t^c)$
$\forall t$
CRF
L^*
Joint training
Raw speech utterance S
CNN
$f_t(S)$
$\frac{\partial f_t(S)}{\partial \theta_f}$
CRF
L^*

(a)

(b)

Figure 7.3 – Illustration of the two training strategies: (a) separate training and (b) joint training.

gradient of the network output $\frac{\partial f_t^{l_t}(S,\theta_f)}{\partial \theta_f}$ is back-propagated to the network.

Separate Training

The networks and the CRF are trained separately. In this strategy, a softmax layer is added to the network to obtain posteriors probabilities $P(l_t|\mathbf{s}_t^c)$ for each speech segment $\mathbf{s}_t^c$ at the output of the network. In this case, the CRF score $c(S, L, \Theta)$ then becomes:

$$c(S, L, \theta_A) = \sum_{t=1}^{T} \left(\log(P(l_t|\mathbf{s}_t^c, \theta_f)) + A_{l_t, l_{t-1}}\right) \tag{7.6}$$

The network parameters θ_f are learned using the cross-entropy criterion, as presented in Section 3.2.2. The likelihood $\mathcal{L}$ is then optimized using (7.5) only with respect to the CRF parameters θ_A:

$$\frac{\partial \mathcal{L}(\Theta)}{\partial \Theta} = \frac{\partial \mathcal{L}(\theta_A)}{\partial \theta_A}. \tag{7.7}$$

Note that using this strategy implies that the acoustic model is trained *locally*, like as in the hybrid HMM-based approach.

7.1.2 Weakly-supervised Training

In weakly-supervised training, we assume that the phoneme segmentation L is not available, only the phoneme transcription $\Lambda = \{\lambda_1, \lambda_2 ..., \lambda_N\}$ of length N is available. The problem of segmentation consists in finding a sequence L (over T frames) of labels, such that aggregation of successive identical labels in L matches the sequence Λ. To infer the segmentation, we need to constrain the CRF graph such that it covers all possible sequences L that could match Λ after label aggregation.

Segmentation Graph

The constraints over time imposed by the label sequence Λ can be written as a directed cyclic graph, where each node represents one label from the sequence, as illustrated in Figure 7.4. At every time step, the path can either stay in the current node through the loop or go to the next node (or label).

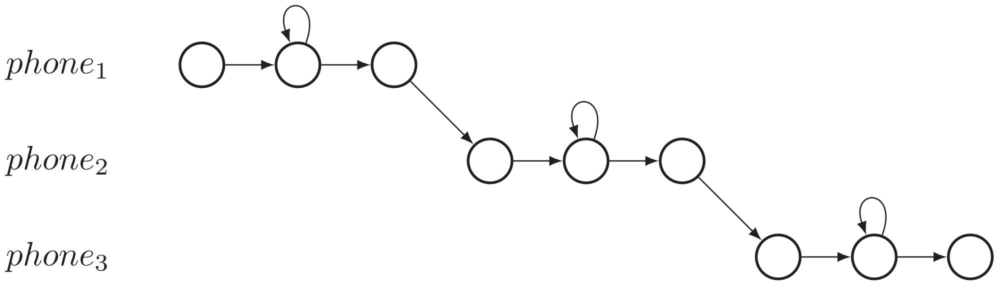


Figure 7.4 – Illustration of the cyclic graph for 3 classes.

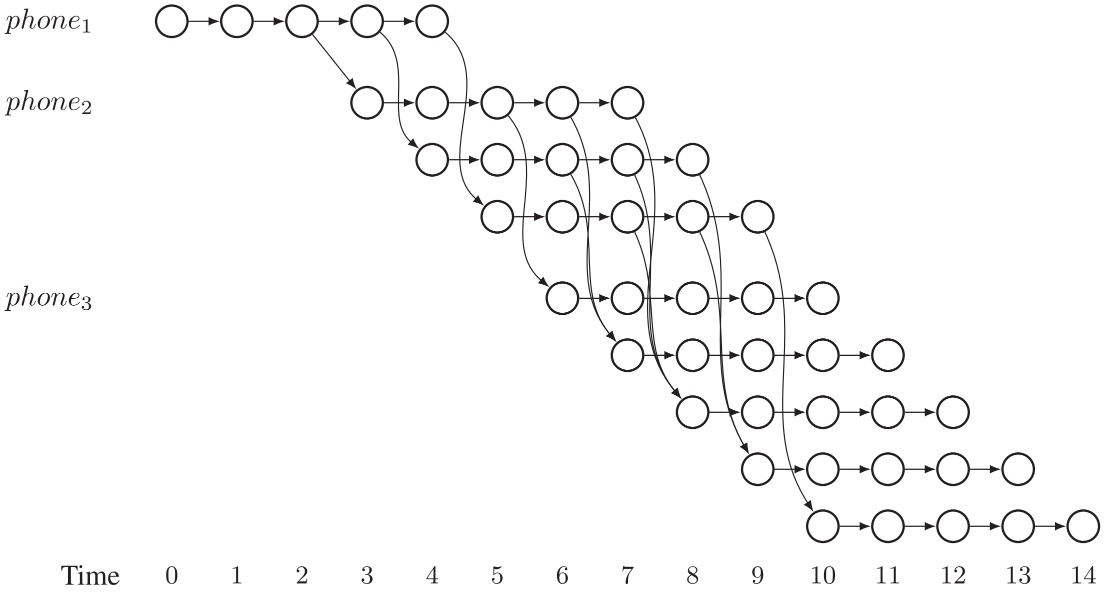


Figure 7.5 – Illustration of the acyclic expanded graph for 3 classes, with $t_{min} = 3$ and $t_{max} = 5$.

In order to implement such graph, we need to expand it to an acyclic graph over a sequence duration of T frames. We introduce two parameters, t_{min} and t_{max}, which represent the minimum and maximum time the path can stay in the same label, expressed in terms of number of frames. To enforce these conditions, the acyclic graph must have multiple parallel branches for each label. All parallel branches of the same label share their weights, i.e. $f_t(S)$ is the same for each time t in each parallel branch. An illustration is provided in Figure 7.5.

In this graph, the number of nodes depends on the length of the phoneme transcription N. The number of parallel branches N_{br} is given by the sum of the parallel branches for each phoneme

$$N_{br} = \sum_{n=1}^{N} (n-1) \cdot (t_{max} - t_{min}) + 1. \tag{7.8}$$

As each branch contains t_{max} nodes, the total number of nodes N_{node} is given by

$$N_{node} = t_{max} \cdot N_{br}. \tag{7.9}$$

For example, for a transcription of length 20 with $t_{max} = 30$ frames and $t_{min} = 3$ frames, the graph has 450k nodes.

Training

In the following, we denote the unconstrained CRF graph over T frames as $\mathcal{U}_T$ (Figure 7.2), and we denote the graph constrained to the right sequence of labels Λ as $\mathcal{C}_T$ (Figure 7.5).

Finding the best sequence L^* ($L^* \subset \mathcal{C}_T$) matching the right sequence of labels Λ sequence corresponds to solving the following maximization problem

$$\max_{C \in \mathcal{C}_T} c(S, C, \theta). \tag{7.10}$$

This is achieved with a Viterbi algorithm, as in (7.2). Specifically, by integrating this best path into (7.4) leads to the following likelihood:

$$\mathcal{L}(\Theta) = \max_{C \in \mathcal{C}_T} c(S, C, \Theta) - \underset{U \in \mathcal{U}_T}{\text{logadd}}\, c(S, U, \Theta). \tag{7.11}$$

The parameters of the network θ_f and of the CRF θ_A are learned jointly by stochastic gradient descent algorithm.

7.2 Phoneme Sequence Recognition Study

In this section we present the experimental setup and the results of the phoneme recognition study on the TIMIT corpus.

7.2.1 Experimental Setup

TIMIT Corpus

The training set, validation set and test set are same as in the previous chapters, detailed in Section 3.3.1. The phoneme set is composed of 61 phonemes. For evaluation, the 61 phonemes are mapped to the 39 phoneme set [Lee and Hon, 1989]. A phoneme segmentation is provided with this corpus. We refer to this segmentation as “manual segmentation”.

CNN-based System Setup

The input features for this part of the study are raw speech waveform, as described in Chapter 3. The architecture is composed of four filter stages. The hyper-parameters are tuned based on the phoneme error rate of the validation set, and are presented in Table 7.1.

Table 7.1 – Network hyper-parameters.

System	# hidden layers	w_{in}	nhu	kW	dW	d_n	kW_{mp}
CNN	1	310 ms	1000	30,7,7,7	5,1,1,1	200,100,100,100	4,2,2,2
	2	310 ms	1000,1000	30,7,7,7	5,1,1,1	200,100,100,100	4,2,2,2
	3	310 ms	1000,1000,1000	30,7,7,7	5,1,1,1	200,100,100,100	4,2,2,2

In the CNN-based architecture, the number of output labels, i.e. the length of the inferred phoneme sequence, is given directly by the hyper-parameters. The duration of one output label T_{lab} (in seconds) is given by the duration of one sample of the input waveform (given by the inverse of the sampling frequency f_s) multiplied by the total pooling N_{pool}, i.e.

$$T_{lab} = \frac{1}{f_s} * N_{pool} \tag{7.12}$$

Using 4 filter stages, the number of pooling is given by:

$$N_{pool} = \prod_{i=1}^{4} dW_i * dW_{mp,i} \tag{7.13}$$

To be consistent with the baselines, the output label duration was set to T_{lab} = 10ms, thus N_{pool} = 160. The hyper-parameters grid search was limited to fit this constraint.

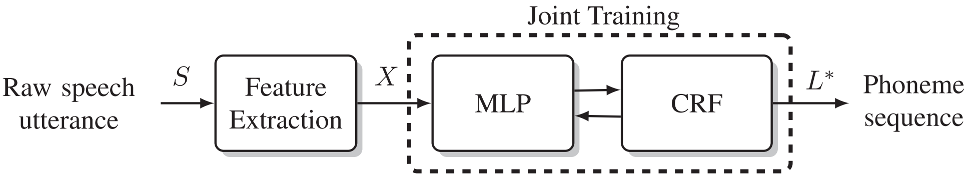


Figure 7.6 – Illustration of the ANN-based system using MFCC features as input.

Baselines

We compare the CNN-based system using raw speech as input to ANN-based systems using MFCC features as inputs. The score for a path in Equation (7.1) becomes:

$$c(X,L,\Theta) = \sum_{t=1}^{T} \left(f_t^{l_t}(X,\theta_f) + A_{l_t,l_{t-1}} \right) \tag{7.14}$$

where $X = \{\mathbf{x}_1 \dots \mathbf{x}_T\}$ is a sequence of feature, as illustrated in Figure 7.6. The system is trained using the three training strategies presented above. We use the same MFCC features as used in the previous TIMIT study in Chapter 3. The classifier is a MLP composed of one to three hidden layers. The number of hidden units for each layer is set to 1000.

For the sake of completeness, we also compare our results to the CRF based system proposed in [Morris and Fosler-Lussier, 2008]. This system uses local posterior estimates provided by an ANN (trained separately using PLP features) as features for the CRF. This system is referred as “CRF”. The second baseline is a ANN/CRF based system [Prabhavalkar and Fosler-Lussier, 2010], where the ANN using PLP features as input is trained jointly with the CRF by back-propagation. It is referred to as “ML-CRF”. All these systems are trained using the 61 phoneme, mapped to the 39 phonemes set for evaluation.

CRF Hyper-parameters

The hyper-parameters of the segmentation graph are the minimum and maximum phoneme duration t_{min} and t_{max}. They are tuned on the phoneme error rate of the validation set. The minimum duration t_{min} was set to 30ms, or 3 frames. The maximum duration t_{max} was set to 300ms, or 30 frames. The maximum duration of the silence class is set to 150 frames, or 1.5 s.

7.2.2 Results

The results on the phoneme sequence recognition task are reported in Table 7.2 for the two training strategies using manual segmentation, namely separate training and joint training, and for the weakly-supervised training strategy. Using manual segmentation, one can see that the ANN-based system with single hidden layer yields similar performance to the CRF baseline (30.2% and 30.7% PER) and to the ML-CRF baseline (29.1% and 28.9% PER). Adding more layer improves the performance. The end-to-end CNN-based system clearly outperforms the CRF baselines and the ANN-based systems. Moreover, the CNN-based system with one hidden layer yields better performance than the ANN-based system using three hidden layers. One can see that the joint training approach leads to similar or better systems than the separate approach.

Systems trained using the weakly-supervised training approach yield similar or better performance than systems trained using manual segmentation. Figure 7.7 illustrates the segmentation obtained by the proposed approach with the manual segmentation for an utterance. It can be observed that there are only minor differences between the segmentations. These results clearly indicate that the proposed weakly supervised training approach, which maximizes $P(L|X)$, can be a good alternative to the independent training approach, based on maximizing $P(L,X)$.

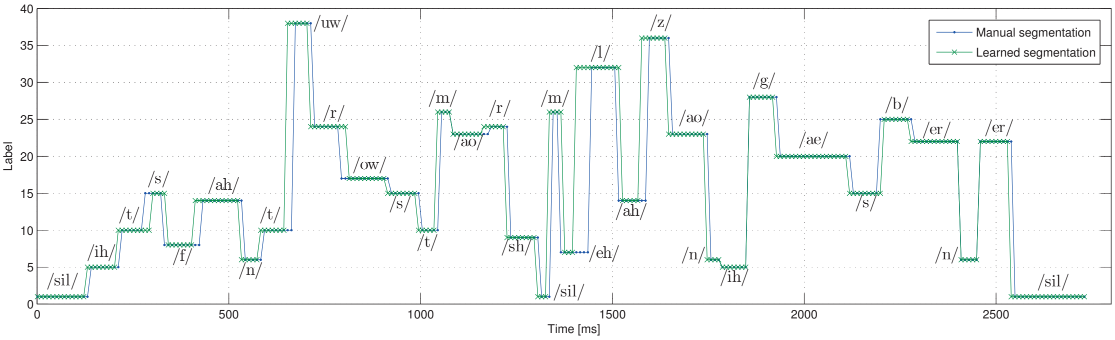


Figure 7.7 – Phoneme segmentation example using the 39 phoneme set, for sequence `sx32` of speaker `mcdc0`.

Table 7.2 – Evaluation of the proposed approach on the TIMIT core testset. Results are expressed in terms of PER. The CRF baseline performance is reported in [Morris and Fosler-Lussier, 2008] and the ML-CRF performance is reported in [Prabhavalkar and Fosler-Lussier, 2010].

Input	Systems	# Hidden Layers	Separate Training	Joint Training	Weakly-sup. Training
Previous works					
MFCC	CRF	1	30.7	-	-
PLP	ML-CRF	1	-	28.9	-
Proposed approach					
MFCC	ANN	1	30.2	29.1	28.7
MFCC	ANN	2	29.9	28.0	27.9
MFCC	ANN	3	29.7	27.6	27.3
RAW	CNN	1	25.6	25.5	26.6
RAW	CNN	2	25.0	25.4	25.7
RAW	CNN	3	24.9	25.4	25.7

7.3 Discussion

In this section, we provide an analysis of the CNN architecture used in this chapter and compare it against the CNN architecture used in Chapter 3. We then contrast the proposed CRF-based approach to the literature.

7.3.1 Analysis and Assessment of the Proposed Approach

In this chapter, we investigated end-to-end training using raw speech as input. Like in Chapter 3, the network hyper-parameters were tuned experimentally on the validation set. The best performance using the end-to-end system was found with a set of hyper-parameters different from the set found with the CNN-based system using HMM-based decoding. The main differences are: (1) the end-to-end system need more filter stages (4 stages) than the hybrid system (3 stages); (2) The first convolution shift is 5 samples, or 0.3 ms, which is shorter than in the hybrid system (10 sample or 0.6 ms); (3) the number of filters in each convolution is higher for the end-to-end system (100 vs 60), specially in the first convolution (200 vs 80). This could be explained by the fact that in the end-to-end system, the CNNs have to model input frames according to the whole utterance, thus the variability of the examples is larger than the hybrid case using limited context.

The performance of the end-to-end system demonstrates the viability of the proposed approach. However, it can be noted that the approach underperforms compared to the hybrid CNN-based system (see Table 3.3). A possible explanation is the estimation of unseen phone transitions. In the HMM-based system, a phone n-gram model is used to decode, which has the in-built capability to handle well unseen phone transitions, e.g. back-off. In the proposed approach, the unseen transition are not handled explicitly and that could be the reason the performance drop. In addition to that in the present study 61 states were used as opposed to 183 states.

7.3.2 Relation to Global Training Methods

End-to-end sequence-to-sequence conversion has been of interest since 1990s. Global training of the acoustic model has been investigated early in the context of hybrid HMM/ANN [Bengio et al., 1991]. The REMAP approach [Bourlard et al., 1994b, 1995] has also been proposed, where $P(W|X)$ is modeled through recursive estimation of static probability conditioned on the current observation and the previous state. Recently, inspired by segment-based approaches [Glass, 2003], segmental CRFs approach [Zweig and Nguyen, 2009] has been proposed for continuous speech recognition task. This approach is based on CRF using segment-level features operating at multiple time scales and language model-level features. Thus, in this approach the acoustic and language models are trained jointly in a discriminative manner. More recently, there has been a growing interest in investigating end-to-end sequence recognition approaches that are able to alleviate the need of pre-segmented labels. The

Connectionist Temporal Classification (CTC) approach [Graves et al., 2006, 2013] has been proposed. It is discussed in the next section. Building on top of that, recurrent models with an attention mechanism have been proposed in speech recognition. Such models are able to select relevant information iteratively. Such approaches have been successfully applied to phoneme recognition [Chorowski et al., 2015] and speech recognition [Chan et al., 2015]. Segmental recurrent neural networks have also been proposed for phoneme recognition [Lu et al., 2016].

Table 7.3 contrasts the proposed approach with above discussed approaches along four dimensions: whether raw speech or spectral based feature is used as input; whether the system is based on frame-level or segment-level classification; whether the acoustic model and the language model (or the phone transition model) are trained jointly and whether the segmentation is obtained by an external system or learned jointly with the system. We can observe that the proposed approach scores positively in three of the four dimensions. However, it is worth noting that segmental CRFs could be used in the proposed approach.

Table 7.3 – Comparison of global training methods. RAW denotes the use of raw speech as input, SA denotes the segment-based classification, JALM denotes the joint training of the acoustic and language model and SL denotes the segmentation learning.

Method	RAW	SA	JALM	SL
Global training of HMM/ANN [Bengio et al., 1991]	✗	✗	✗	✗
REMAP [Bourlard et al., 1994b]	✗	✗	✓	✓
Segmental CRFs [Zweig and Nguyen, 2009]	✗	✓	✓	✗
Connectionist Temporal Classification (CTC) [Graves et al., 2013]	✗	✗	✗	✓
Proposed CRF	✓	✗	✓	✓
Attention-based Models [Chorowski et al., 2015]	✗	✗	✓	✓
Segmental RNN [Lu et al., 2016]	✗	✓	✓	✓

7.3.3 Relation to Connectionist Temporal Classification

The Connectionist Temporal Classification approach [Graves et al., 2006] proposed a method for labeling sequences without the need for pre-segmented data. More specifically, this approach is presented as a method to train RNN-based acoustic model. The training criterion is based on maximizing the conditional probability of the correct phoneme sequence given the input sequence. This approach, similar to our approach, is able to learn the segmentation jointly with the acoustic modeling. The key differences between the CTC approach and our approach are the following:

1. The CTC approach does not model the phoneme transition.
2. In the CTC approach, the output of the network is constrained to be posterior probabilities by using a softmax layer. We do not use such constraint.

3. The acoustic model in the CTC approach is a recurrent neural network. Thus, the time dependence between successive acoustic observations are modeled explicitly in the network, where in our approach, it is modeled implicitly, as the NN-based model is trained through the CRF.

4. In terms of performance, the best result reported by the first CTC-based study [Graves et al., 2006] on the TIMIT corpus is 30.1 % PER, which is worse than the performance of our approach. This performance is evaluated on the full test set, known to be easier that the core test set used in this thesis. This shows that our approach can be a good alternative to CTC.

5. Recently, the CTC approach was used in the context on BLSTM-based transducers [Graves et al., 2013]. This system yields state-of-the-art performance on TIMIT core testset (17.4% PER). In this case, the phoneme transitions are modeled independently.

7.3.4 Relation to Sequence-discriminative Approaches for Acoustic Modeling

In the proposed approach, the models are trained by emphasizing the score of the true sequence while de-emphasizing the score of all other or competing sequences. In that sense, the proposed approach can be seen as similar to the sequence-discriminative training framework, which uses criteria inspired from HMM/GMM systems [Gales and Young, 2007], like Maximum Mutual Information (MMI), state Minimum Bayesian Risk (sMBR) or Minimum Phone Error (MPE) [Kingsbury, 2009, Guangsen and Sim, 2011, Andrew and Bilmes, 2012, Vesely et al., 2013] and thus could have potential implications for discriminative acoustic modeling . The key difference between the two approaches is that in [Vesely et al., 2013] sequence discriminative training is done in several steps. More precisely, training of a local ANN (or deep neural network) classifier with cross entropy criterion followed by sequence discriminative training of the ANN using a cost function based on maximum mutual information [Bahl et al., 1986] or minimum phone error [Povey and Woodland, 2002] criteria. In the proposed approach, as described earlier, there is no intermediate local classifier training. All the parameters are trained in end-to-end manner based on sequence discriminative error criteria. The other difference lies in the implementation of sequence discrimination criteria. In the MMI or MPE case, the score normalization is done by summing over all possible word hypotheses, which is practically infeasible to estimate. So it is approximated by decoding the training data using a bigram or trigram language model and generating a lattice. In our case, it is computed by using a fully connected phone model, which can encompass the phone state sequences corresponding to all possible word sequences.

Thus, the proposed approach could alternately be used to estimate sequence-discriminative local phone posterior probabilities given the global input signal. Indeed this can be done by using forward-backward algorithm in CRF [Lafferty et al., 2001, Fosler and Morris, 2008].

7.4 Summary

In this chapter, we proposed a sequence-to-sequence conversion approach which takes raw speech utterance as input and outputs a phoneme sequence. The system is trained in an end-to-end manner, where every step is trained jointly with the others. We also presented a weakly-supervised training strategy, where the system learns the phoneme segmentation from the transcription. We showed that use of raw speech as input to a CNN yields better system than ANN-based system using cepstral feature as input, which is consistent with the findings in Chapter 3.

8 Jointly Learning to Locate and Classify Words

In the previous chapter, we investigated a sequence-to-sequence conversion approach, which takes a speech utterance as input and outputs a phoneme sequence. We showed that such a system can be trained in a weakly-supervised manner, where only the phoneme transcription is needed for the training. In this chapter, we investigate relaxing the label sequence ordering. In other words, we discard the sequence information at the output of the system and treat the sequence-to-sequence prediction problem as a multi-label classification problem.

Specifically, we propose a novel multi-word detection system. The system is composed of two stages: a sequence modeling stage, based on convolutional neural networks, which performs the acoustic modeling and outputs a score for each frame, for each word. The second stage is the aggregation stage, which aggregates the score computed by the CNNs along the temporal dimension. The system is trained using bag-of-word as label, which denotes the presence information of words in a speech utterance, and is able to *learn* the words localization and classification jointly.

8.1 Related Work

There is a growing interest in applying the deep learning approach to weakly-supervised systems. At the time of training, these pattern recognition systems have only access to the “presence or absence” information of a pattern in a given input, and learn which part of the input is relevant for classifying the pattern. In computer vision, this approach has been successfully applied to image segmentation [Pinheiro and Collobert, 2015]. Attention-based recurrent models have also been developed recently, which iteratively process their input by selecting relevant information at every step. They have been successfully applied to handwriting synthesis [Graves, 2013], visual object classification [Mnih et al., 2014] and machine translation [Bahdanau et al., 2014]. Recently, such an approach has been applied to phoneme recognition task [Chorowski et al., 2015] and yields state-of-the-art performance while being able to infer phoneme segmentation. In these approach however, it was always assumed that either the segmentation of the training data or at least the sequence information (order of the

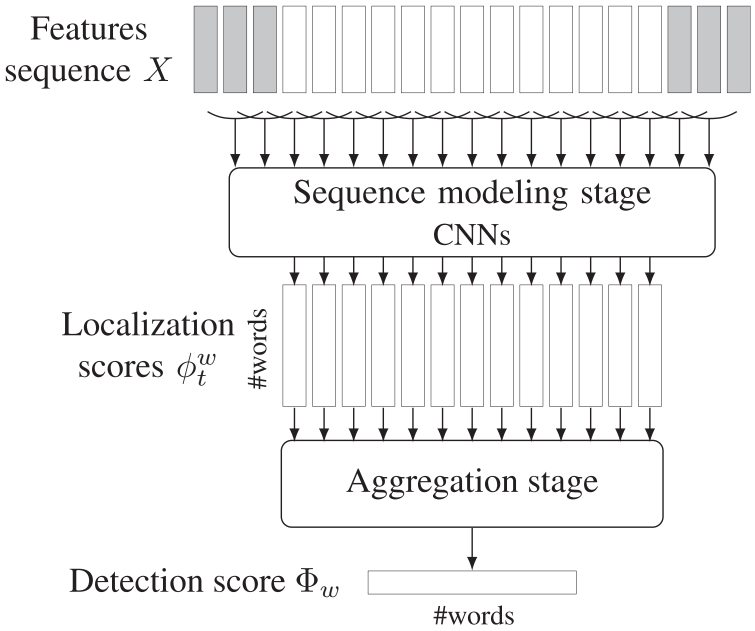


Figure 8.1 – Illustration of the proposed system. The gray input frames represent the padding.

words) is provided. The proposed approach does not make such an assumption.

8.2 Proposed Approach

The proposed approach takes a feature sequence X as input, and outputs the probability of each word w in the dictionary $\mathcal{D}$ being present in the utterance. During training, the targets are Bag-of-Word labels, which is a binary vector denoting the presence or absence information of words in the utterance.

8.2.1 Two-stage CNN-based System

Figure 8.1 presents the proposed system which is composed of two stages: the *sequence modeling stage* processes a sequence of features and outputs a score for each word at each frame. The *aggregation stage* performs the aggregation of the scores along the temporal dimension and outputs a score for each word for the whole utterance. Both stages are trained jointly.

Sequence Modeling Stage

The sequence modeling stage models the acoustic sequence. More precisely, the network is given a sequence of features $X = [\mathbf{x}_1\ \mathbf{x}_2\ \dots\ \mathbf{x}_T]$, where $\mathbf{x}_t$ stands for a feature vector at time frame t. The output is a score $\phi_t^w(X)$ for each frame t and each word $w \in \mathcal{D}$. This score is referred to as the *localisation score*.

This stage is implemented by a succession of N convolution layers. A convolutional layer applies the same transformation over each successive (or interspaced by dW frames) windows of kW frames, as presented in Section 3.2.1. In this chapter, we refer to the convolution layer operation on input X followed by a non-linearity as $Conv(X)$. The localisation score can thus be expressed as:

$$\phi_t^w(X) = Conv_N(Conv_{N-1}(...Conv_1(X))) \tag{8.1}$$

Aggregation Stage

For a given sequence X of length T, the sequence modeling stage produces a score $\phi_t^w(X)$ for each frame t and each word $w \in \mathcal{D}$. Given that at the training time we have only access to the bag-of-word labels, we need a way to aggregate these frame-level scores into a single sequence-level score $\Phi_w = aggreg(\phi_t^w)$, referred to as the *detection* score.

The aggregation $aggreg(\cdot)$ should drive the network towards correct frame-level assignments. A possible aggregation would be to take the sum over all frames: $\Phi_w = \sum_t \phi_t^w(X)$. This would however assign the same weight on all frames of the speech sequence during the training procedure, even to the ones which do not belong to the words corresponding to the labels. On the other hand, one could apply a max aggregation: $\Phi_w = \max_t(\phi_t^w)$. This would encourage the model to increase the score of the frame which is considered as the most important for the classification of a given word. With this approach, the position of a given word would be correctly predicted, but its duration would not, as only one frame is encouraged. We propose a trade-off solution between these two cases, which is the `LogSumExp` [Boyd and Vandenberghe, 2004] (LSE):

$$\Phi_w^r(X) = \frac{1}{r} \log\left(\frac{1}{T}\sum_{t=1}^{T} \exp(r \cdot \phi_t^w(X))\right) \tag{8.2}$$

where r denotes the hyper-parameter controlling how smooth one wants the approximation to be: high r value ($r \gg 1$) implies having an effect similar to the max, very low value ($r \ll 1$) will have an effect similar to the score averaging. The advantage of this aggregation is that the frames which have similar scores will have a similar weight in the training procedure.

8.2.2 Training

Bag-of-word Labels

As mentioned previously, we use *Bag-of-words* (BoW) labels. Based on the bag-of-word model used in natural language processing, for a given utterance these labels denote the “presence or absence” information of each word in the dictionary. They are extracted from the transcription, and are represented by a binary vector **y**, of dimension equal to the dictionary size. Note that such labels neither take into account the words order nor quantity. For example, given the transcription "*John likes to watch movies. Mary enjoys movies too.*", the resulting BoW labels are: {"enjoys","likes", "movies","to", "too","watch" }, assuming that "John" and "Mary" are not in the dictionary. The binary label vector for this utterance can then be built by setting to 1 the entries corresponding to the indices of the words and −1 all the other entries of the dictionary.

Cost Function

As more than one word can be present in an utterance, the standard cross-entropy cost function is not suited in this case. We propose to treat the task as a separate binary classification problem for each word. Given the bag-of-word label $\mathbf{y} = [y_1 \dots y_m \dots y_{|\mathcal{D}|}]$, with $y_m \in \{-1, 1\}$, denoting the presence or absence of the word w in the input utterance X, the cost function $\mathcal{L}$ is thus a sum of of $|\mathcal{D}|$ binary logistic regression classifiers:

$$\mathcal{L}(\Phi(X), \mathbf{y}) = \sum_{w=1}^{|\mathcal{D}|} \log(1 + e^{-y_w \Phi_w(X)}) \tag{8.3}$$

with $\Phi_w(x)$ being the detection score for the word w. Treating a multi-label classification problem as a sum of independent classifiers may seem to be inadequate, but in our approach, the binary classifiers are not totally independent as they share hidden layers (in the sequence modeling stage), which could model the inter-label dependencies, if any.

8.2.3 Inference

During inference, the unseen utterance X is given as input to the system. The system will produce as output the detection score $\Phi_w(X)$ (as defined in Equation (8.2)) for each word in the dictionary. Using this score, the probability $P(w|X)$ of the word w being present in the utterance can be computed as:

$$P(w|X) = \frac{1}{1 + e^{-\Phi_w(X)}} \tag{8.4}$$

This probability can be used to decide presence of absence of the word w in the utterance. In some cases, the detection information alone is not enough, for instance when the word localisation information is required. We assume that, for a given word, the localisation score ϕ_t^w is a measure of the likelihood of the word being in the utterance at time t. Based on that assumption, the most likely position pos_w of a given word, i.e. the most probable frame, can be computed as:

$$pos_w = \underset{t}{\operatorname{argmax}}(\phi_t^w) \tag{8.5}$$

In order to obtain the duration of a given word, a simple model is proposed: a threshold is applied to the localisation score for the given word. Thus, the word localisation is given by each frame whose scores are higher than the threshold. A threshold per word is used, and is determined experimentally,

$$\phi_t^w > \theta_w, \quad \forall t, \tag{8.6}$$

with θ_w being the threshold for the word w. Note that it is possible to detect more than one occurrence of a given word in the utterance with this method.

8.3 Experimental Setup

In this section, we present the database, the setup of the proposed system and the studies, namely, the word localization study and the keyword spotting study.

8.3.1 Database

The LibriSpeech corpus [Panayotov et al., 2015] is an English corpus derived from read audio books, sampled at 16 kHz, The trainset consists of 280k utterances, representing 960 hours of speech. Two development and test sets are available. In both cases, the first set is composed of high quality utterances (i.e. having the lowest WER when recognized by ASR system) and is referred to as *dev_clean* and *test_clean*. The second one is composed of low quality utterances (i.e. having the highest WER when recognized by ASR systems), and referred to as *dev_other* and *test_other*. Each of these sets consists of 40 speakers, and represents about 5 hours of speech. To obtain the word alignments, we use the `s5` recipe, provided by the Kaldi toolbox [Povey et al., 2011]. It is a HMM/GMM system using MFCCs; more details can be found in [Panayotov et al., 2015].

8.3.2 Proposed System Setup

To demonstrate the viability of the proposed approach, we use Mel Filterbanks energies as input features instead of raw speech signal as used in the previous Chapters. These features were computed using the `Spectral` package[1]. They consist of 40 coefficients, computed on a 25 ms window, with a 10 ms shift, without any temporal derivatives. The hyper-parameters of the network were tuned on the validation set by maximizing the F1 score. In the results, we used a detection probability threshold of 0.4, that yields a F1 score (on words) of 0.72 on the clean development set, and 0.6 on the other development set. The proposed architecture is composed of 10 convolutions layers. The first layer has a kernel width kW of 5 frames, the 9 other layers have a kernel width of 10 frames. They all have a shift dW of 1 frame, and 80 filters. The dictionary $\mathcal{D}$ consist of 1000 most common words in the training set. The BoW target were based on that dictionary. We train the network using stochastic gradient descent [Bottou, 1991] with a learning rate of 10^{-5}. The experiments were implemented using the Torch7 toolbox.

---

[1] https://github.com/mwv/spectral

8.3.3 Evaluation Studies

We present here the details of the word localization study and the keyword spotting study that are used to demonstrate the potential of the proposed approach.

Word Localisation Study

To evaluate the capability of the proposed approach to *learn* the word localisation in a weakly-supervised manner, we conducted two studies,

1. **Word position study**: In this study, we first evaluate the capability of the system to detect the correct word position in an utterance in the following manner. For each utterance, the most probable position of a given word is computed using Equation (8.5). We then check if this position is correct (i.e. if the word is present at this frame on the ground-truth labels). We propose two evaluation settings. In the first one, referred to as *oracle*, the word detection capability of the system is assumed to be perfect, i.e. we use the ground-truth BoW labels to detect words in utterance. In the second setup, referred to as *actual*, we perform a word detection by thresholding the probability of the word being present in the sequence using (8.4), and then compute the position accuracy as presented above. In this case, the threshold was tuned to maximize the F1 score on word classification on the validation set.

2. **Word duration study**: In this study, we evaluate the system's capabilities to predict the correct word duration. As presented in Section 8.2.3, the duration of a given word is inferred by thresholding the localisation score. For evaluation, we use the Intersection-over-Union (IoU) metric. This metric can be seen as a proximity measure between two patterns, as it is equal to 0 if they do not overlap, and equal to 1 if they are perfectly matching. A IoU score of 0.5 indicates that half of the patterns match. It is well used for image segmentation (see [Pinheiro and Collobert, 2015] for example). Formally, it is defined as:

   $$U^{(w)}_{\text{iou}}(\tilde{L}, L) = \frac{\sum_t \mathbb{1}_{\{\tilde{l}_t = w \wedge l_t = w\}}}{\sum_t \mathbb{1}_{\{\tilde{l}_t = w \vee l_t = w\}}} \tag{8.7}$$

   with $\tilde{L} = \{\tilde{l}_1 \dots \tilde{l}_T\}$ denotes the inferred sequence, $L = \{l_1 \dots l_T\}$ denotes the reference, $w \in \mathcal{D}$ denotes a given word and $\mathbb{1}_{\{predicate\}}$ denotes the indicator function, which is 1 if the predicate is true and 0 otherwise.

For these two studies, we use the frame-level word alignment obtained by the HMM/GMM system as ground-truth.

Keywords Spotting Study

To demonstrate a real word application of the proposed approach, we present a keyword spotting study, where

1. The keywords spotted are in-vocabulary words, i.e. words seen during training.
2. As mentioned in Section 8.3, the word dictionary is limited to the 1000 most common words in the corpus. Thus, the keywords selected for the study are part of this subset. This is unusual for KWS studies, as the selected keywords are usually quite uncommon. This constraint is selected for practical reasons, mainly for training speed. However, the number of words in the dictionary is a hyper-parameter, and could be extended to any number of words.

The set of keywords used is presented in Table 8.1.

Table 8.1 – Keywords list (in vocabulary).

any	battle	birds	cannot
easily	fifty	filled	great
known	land	lie	never
only	perfect	perhaps	presence
show	thank	them	years

For evaluation, we used the Maximum Term Weight Value (MTWV) metric as expressed in Equation (2.31) with the number of trial per second $n_{pr} = 1$, the cost over value ratio $C/V = 0.1$ and the term prior probability $P_{tr} = 10^{-4}$, as presented in Section 2.6. We used the F4DE tool [f4d] provided by NIST for scoring.

Proposed approach To detect and localize keywords with the proposed system, we use the following procedure. For each utterance, the presence of keyword is determined by thresholding the probability $P(w|X)$ as defined in Equation (8.4). The starting and ending time stamps of the keyword are then computed by thresholding the localisation score, as presented in Equation (8.6).

Baseline We use the LVCSR lattice-based KWS system provided with the Kaldi toolbox[2] as baseline.

---

[2] http://kaldi.sourceforge.net/kws.html

8.4 Results

In this section, we present the results for the word localization study and the keyword spotting study.

8.4.1 Word localisation study

The results for the word position study are presented in Table 8.2 in terms of position accuracies, for the *oracle* setup and for the *actual* setup. Using the *oracle* setup, where the detection capability of the system is perfect, one can see that the proposed system is able to correctly detect the position of most of the word occurrences in the test sets. In the *actual* setup, the result indicates that more than half of word occurrence are correctly *detected* and correctly *localized*.

Table 8.2 – Word position accuracies.

Set	Oracle	Actual
test_clean	87.1 %	60.1 %
test_other	83.5 %	55.2 %

Figure 8.2 presents the results for the word duration study, in term of mean IoU for each word in the dictionary. One can see that on average, about one third of the word duration is captured. Figure 8.3 presents an illustration of an inferred sequence and the ground-truth. Unsurprisingly, the proposed system predicts shorter duration.

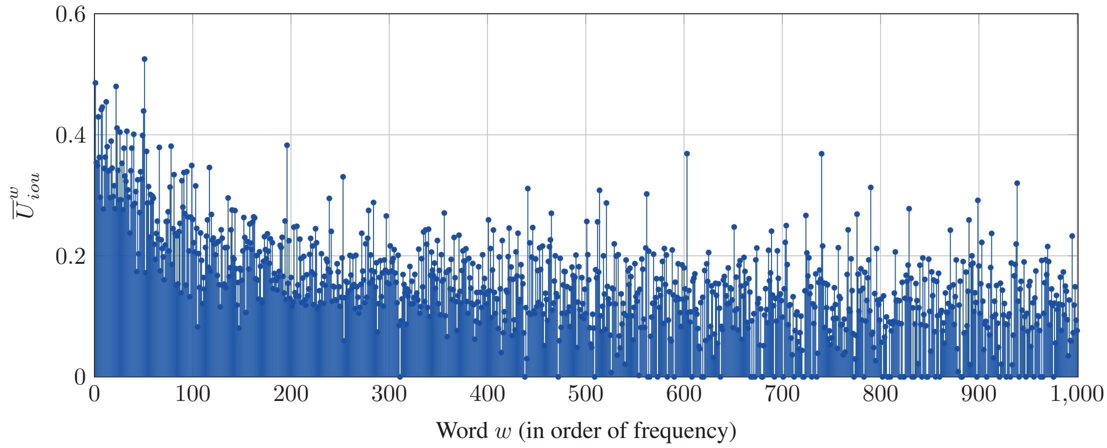


Figure 8.2 – Mean IoU for each word on the *test_clean* set.

8.4.2 Keywords Spotting Study

Table 8.3 presents the results for the keyword spotting study for the proposed system and the baseline system, expressed in terms of MTWV. On the *test_clean*, the proposed system yields

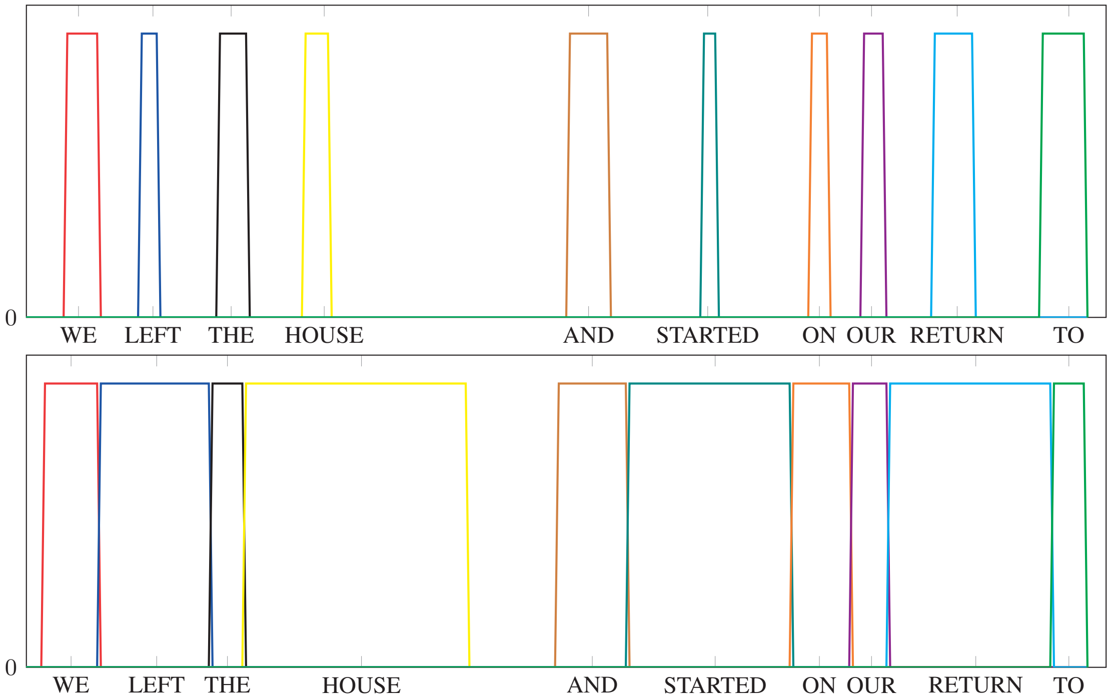


Figure 8.3 – Illustration of an inferred sequence on the top and its corresponding ground-truth, on the bottom.

similar results to the baseline. This result clearly indicates that the proposed system is able to jointly learn to localize and classify words. On the *test_other* set, the performance gap between the proposed system and the baseline suggests that the proposed system is less robust than the baseline under mismatched conditions.

Table 8.3 – Keyword spotting performance on the *test_clean* and the *test_clean* set of LibriSpeech.

Set	System	MTWV
test_clean	Baseline	0.72
	Proposed	0.69
test_other	Baseline	0.49
	Proposed	0.33

8.5 Analysis

Our studies demonstrated that the proposed approach is able to learn word localisation in a weakly-supervised manner and could yield performance similar to the baseline system on keyword spotting task. S question that arises is: what have the networks learned?

In the proposed architecture, the word classification and localisation is performed by the layer just before the aggregation, i.e. the layer which computes the localisation score s_t^w, defined in Equation (8.1). This score s_t^w for a given word w_i can be seen as the dot product between the i^{th} row of the weight matrix and the sequence representation computed by the previous layer. Thus, each row can be seen as a vector representation of a given word. In the literature, this kind of word representation are often referred as *word embedding*, mainly used in natural language processing [Collobert and Weston, 2008]. In speech recognition, such approach has been successfully investigated by [Bengio and Heigold, 2014].

To gain insights on these embeddings, we examined the nearest neighbors in terms of Euclidean distance of the embeddings. Table 8.4 presents the 10 nearest neighbors for six selected examples. It can be observed that most of the neighbor sound similar to the reference words. Alternately, the learned embeddings seems to capture the *acoustic* similarity between words. Leveraging these embeddings is open for further research.

Table 8.4 – Nearest neighbors examples (in column).

place	own	way	drawn	marry	beginning
places	old	away	grown	mary	dinner
face	hold	wait	strong	married	begin
placed	whole	lay	brought	marriage	come
french	beautiful	laid	upon	american	began
prince	almost	later	bright	very	again
race	lower	late	broad	land	didnt
pleased	home	length	son	large	get
case	fellow	work	cause	learned	given
raised	rather	lady	sun	with	doing
grace	arm	word	trying	man	happened

8.6 Summary

We presented a novel approach to jointly localize and classify words based on CNNs. The proposed approach is trained in a weakly-supervised manner, using bag-of-words labels. We demonstrated that the proposed system is able to learn to localize and classify word jointly and could yield a keyword spotting system competitive to standard lattice-based search system.

9 Conclusions

This thesis was devoted towards the development of end-to-end speech recognition systems. To this aim, our research focussed along two main directions: learning the features and the classifier jointly for acoustic modeling and joint modeling of the acoustic model and the sequence decoder. In this thesis, Chapters 3 to 6 were devoted to the first research direction, i.e. end-to-end acoustic modeling. Chapter 7 presented an end-to-end sequence to sequence conversion approach for phoneme sequence recognition. Finally, Chapter 8 investigated a weakly-supervised word localization and recognition system.

In Chapter 3, we investigated a novel CNN-based acoustic modeling approach that automatically learns relevant representations from the speech signal and estimates phone class conditional probabilities for ASR. In this approach, the acoustic model consists of a feature stage and a classifier stage which are jointly learned during training. Specifically, the input to the acoustic model is raw speech signal, which is processed by several convolution layers (feature stage) and classified by an MLP (classifier stage) to estimate phone class conditional probabilities. We evaluated the approach against the conventional acoustic modeling approach, which consists of independent steps: short-term spectral based feature extraction and classifier training. Phone recognition studies on English and ASR studies on multiple languages (English, French, German) showed that the proposed acoustic modeling approach can yield better recognition systems.

In Chapter 4, we presented an analysis of the CNN-based approach using raw speech as input. The proposed analysis was undertaken at two levels: we first analyzed the first convolution layer and then the intermediate features, i.e. the features learned by the feature learning stage. The key findings of the first convolution layer analysis are the following:

1. Both the conventional acoustic modeling approach and the proposed approach tend to model spectral information present in time span of about 200 ms for phone classification. However, they differ in the manner analysis is performed over that time span and feature representations are obtained. Indeed in the proposed approach, contrary to the conventional wisdom of short-term processing, the signal is processed at sub-segmental

level (speech signal of about 2 ms) by the first convolution layer. The subsequent convolution layers temporally filter and integrate the output of first convolution layer to yield an intermediate representation. In other words, the intermediate representation is obtained by processing the information present in the sub-segmental speech signal at multiple temporal resolutions.

2. The filters in the first convolution layer learn from the sub-segmental speech signal a dictionary of matched filters that discriminate phones. Specifically, these filters were found to model formant-like information in the spectral envelop of the sub-segmental speech. These findings are particularly interesting. First, it validates the notion of formants and phone discrimination in a data-driven manner, i.e. without making any explicit assumption about the speech production model. Secondly, sub-segmental spectral processing means high time resolution and low frequency resolution. Conventional method of short-term processing (i.e. determination of the window size) has been developed considering the trade-off between time resolution and frequency resolution. Our investigations show that loss of frequency resolution due to sub-segmental speech processing is not affecting the ASR performance.

The intermediate feature representation analysis led to the following insights:

1. The representations have some level of invariance across domains and languages. More specifically, we observed that the variation of the learned features seems to come more from the domain characteristics as opposed to the set of subword units from the languages. This indicates that learning features in a data-driven manner could lead to language-independent features, like the standard cepstral-based features.
2. These learned representations are more discriminative than standard cepstral-based features. This observation confirms the hypothesis that learning the features and the classifiers jointly leads to more optimal systems (compared to standard “divide and conquer” approaches).

In Chapter 5, motivated by the findings of the discriminative features study, we further investigated the CNN-based approach, where the feature stage has a deep architecture and the classifier has a shallow architecture. We showed that the proposed CNN-based approach allows shifting of the capacity of the system from the classifier to the feature stage with little or no drop in performance. We applied this approach realistically on continuous speech recognition task to demonstrate that it can indeed result in a system that is as efficient as standard HMM/ANN-based system using cepstral features or CNN-based approach with on hidden layer in terms of ASR performance while drastically reducing the capacity of the system.

Learning the feature automatically from the raw speech signal raises the issue of noise robustness of such system. In Chapter 6, we studied the noise robustness of the CNN system using raw speech as input. We presented a robust CNN-based approach where the intermediate representation are normalized in an online manner. This approach was shown to outperform baseline systems using normalized cepstral-based features as input. We also showed that the CNN-based system can be robust in multi-conditional training setup without the normalization technique, unlike the cepstral-based feature based systems, which systematically gain from the cepstral mean and variance normalization technique. We also studied enhancement of speech using ETSI AFE before being fed into the CNN. The results did not show any clear benefits. Whether speech enhancement would really help the proposed CNN-based approach is open for further research.

In Chapter 7, we proposed a novel phoneme sequence-to-sequence conversion model which takes raw speech sequence as input and outputs a phoneme sequence. The system is trained in an end-to-end manner using a weakly-supervised training approach, where the system is able to learn the phoneme segmentation jointly with the phoneme sequence prediction. We showed that this approach yields similar or better performance than baseline systems trained using manual segmentation. This study demonstrated the viability of the proposed approach.

Finally, in Chapter 8, we proposed a word detection system, trained in a weakly-supervised manner using bag-of-word label representation of training utterances. Our studies demonstrate the viability of the weakly-supervised approach for word detection and localization. It could be a first step towards the development of weakly-supervised ASR systems through exploitation of partly labeled data.

In conclusion, this thesis showed that:

1. Feature relevant for ASR can be automatically learned from the speech signal and better systems can be developed using end-to-end acoustic modeling.
2. Weakly-supervised sequence-to-sequence conversion is a viable alternative to the standard ASR approach.

9.1 Direction of Future Research

Raw speech based system The proposed raw speech-based system could be improved along the following directions:

- The features learned by the CNN from the raw speech have been shown to have some level of invariance across languages. One possible approach to increase the robustness could be multi-lingual training, where the system is trained using several languages. This can be achieved using multi-lingual phone set. Multi-task training approach [Caruana, 1997] could also be considered, where the filter stages are shared across languages and the classifier is unique for each language.
- We have observed that the feature stage has considerably fewer parameters than the classifier stage. This provides new means to rapidly adapt the acoustic model. Specifically,

one of the main challenge often faced in adapting the acoustic model to new domains is the amount of adaptation data available. The data may not be sufficient to effectively adapt all the parameters in the acoustic model. In the proposed approach, this challenge could be addressed by only adapting the feature stage. Such an approach would be analogous to maximum likelihood linear regression (MLLR) adaptation approach [Gales and Woodland, 1996] where MLLR is used to transform the features as opposed to the models (i.e. means and variances of the Gaussians). However, in comparison to that, adaptation in the proposed framework would present two distinctive advantages. First, the adaptation would by default be discriminative, i.e. learned by improving discrimination between the phone classes. Second, upon availability of more data adaptation of both feature stage and classifier stage could be effectively employed.

- As the proposed approach makes minimal assumptions and uses minimal prior on the data, the raw speech-based approach could be considered in other speech processing applications, such as speaker recognition or emotion recognition. In that respect, it is worth mentioning that inspired by our end-to-end acoustic modeling approach, multi-channel acoustic modeling [Hoshen et al., 2015] and end-to-end emotion recognition [Trigeorgis et al., 2016] approaches have been proposed.

End-to-end sequence recognition system In this thesis, we presented a phoneme recognition study to study the viability of the the proposed CRF-based approach for end-to-end sequence conversion. Extending this approach to continuous speech recognition is an open problem that pose several great challenges, such as the language model estimation. In the standard HMM-based approach, the language model is estimated independently, usually on a large text corpora. In the proposed approach, the language model would be estimated jointly with the acoustic model, on the same data. Thus, it implies having a dataset suitable for both tasks. This is a highly challenging problem and is an up-and-coming research direction [Graves and Jaitly, 2014, Amodei et al., 2015].

Weakly-supervised multi-word detection system The weakly-supervised multi-word detection system could be extended to a continuous speech recognition system, by adding a decoder. Also, a limitation of the proposed approach on keyword spotting task is that keywords have to be in the dictionary used during training. To address the issue of out-of-vocabulary keyword spotting, one possible approach could be to take advantages of the word embeddings learned by the system. For example, an approach based on generating proxy embeddings, i.e use of embeddings of acoustically close words could be considered.

Bibliography

F4DE NIST tools. http://www.itl.nist.gov/iad/mig/tools/.

O. Abdel-Hamid and H. Jiang. Rapid and effective speaker adaptation of convolutional neural network based models for speech recognition. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 1248–1252, 2013.

O. Abdel-Hamid, A. Mohamed, H. Jiang, and G. Penn. Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4277–4280, 2012.

O. Abdel-Hamid, L. Deng, D. Yu, and H. Jiang. Deep segmental neural networks for speech recognition. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 1849–1853, 2013.

J.B. Allen and L. Rabiner. A unified approach to short-time Fourier analysis and synthesis. *Proceedings of the IEEE*, 65(11):1558–1564, Nov 1977.

D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. C. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos, E. Elsen, J. Engel, L. Fan, C. Fougner, T. Han, A. Y. Hannun, B. Jun, P. LeGresley, L. Lin, S. Narang, A. Y. Ng, S. Ozair, R. Prenger, J. Raiman, S. Satheesh, D. Seetapun, S. Sengupta, Y. Wang, Z. Wang, C. Wang, B. Xiao, D. Yogatama, J. Zhan, and Z. Zhu. Deep Speech 2: End-to-End Speech Recognition in English and Mandarin. *arXiv preprint arXiv:1512.02595*, 2015.

G. Andrew and J. Bilmes. Sequential Deep Belief Networks. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4265–4268, March 2012.

J. Ba and R. Caruana. Do deep nets really need to be deep? In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, pages 2654–2662, 2014.

D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representation (ICLR)*, 2014.

L. R. Bahl, F. Jelinek, and R. L. Mercer. A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (2):179–190, 1983.

L. R. Bahl, P. F. Brown, P. V. De Souza, and R. L. Mercer. Maximum mutual information estimation of hidden Markov model parameters for speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 86, pages 49–52, 1986.

L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The annals of mathematical statistics*, 41(1):164–171, 1970.

S. Bengio and G. Heigold. Word Embeddings for Speech Recognition. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 1053–1057, 2014.

Y. Bengio. A Connectionist Approach to Speech Recognition. *International Journal on Pattern Recognition and Artificial Intelligence*, 7(4):647–668, 1993.

Y. Bengio, R. De Mori, G. Flammia, and R. Kompe. Global optimization of a neural network-hidden Markov model hybrid. In *Proc. of IJCNN*, volume ii, pages 789 –794 vol.2, jul 1991.

Y. Bengio, P. Lamblin, Da. Popovici, H. Larochelle, et al. Greedy layer-wise training of deep networks. *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, 19:153, 2007.

G. Bernardis and H. Bourlard. Improving posterior based confidence measures in hybrid HMM/ANN speech recognition systems. In *Proceedings of International Conference on Spoken Language Processing (ICSLP) Sydney, Australia*, pages 775–778, 1998.

E. Bocchieri and D. Dimitriadis. Investigating deep neural network based transforms of robust audio features for LVCSR. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6709–6713, 2013.

S. F. Boll. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 27(2):113–120, 1979.

L. Bottou. Stochastic Gradient Learning in Neural Networks. In *Proceedings of Neuro-Nîmes 91*, Nimes, France, 1991. EC2.

L. Bottou, F. Fogelman Soulié, P. Blanchet, and J. S. Lienard. Experiments with Time Delay Networks and Dynamic Time Warping for Speaker Independent Isolated Digit Recognition. In *Proceedings of the European Conference on Speech Communication and Technology (EUROSPEECH)*, volume 2, pages 537–540, Paris, France, 1989.

L. Bottou, Y. Bengio, and Y. LeCun. Global Training of Document Processing Systems using Graph Transformer Networks. In *Proceedings of Computer Vision and Pattern Recognition*, pages 490–494. Puerto-Rico., 1997.

H. Bourlard and S. Dupont. Subband-based speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 2, pages 1251–1254. IEEE, 1997.

H. Bourlard and N. Morgan. *Connectionist speech recognition: a hybrid approach*, volume 247. Springer, 1994.

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

H. Bourlard, S. Dupont, and C. Ris. Multi-stream speech recognition. Technical report.

H. Bourlard, B. D'hoore, and J.-M. Boite. Optimizing recognition and rejection performance in wordspotting systems. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, pages I–373. IEEE, 1994a.

H. Bourlard, Y. Konig, and N. Morgan. *REMAP: Recursive Estimation and Maximization of a Posteriori Probabilities; Application to Transition-based Connectionist Speech Recognition*. ICSI, 1994b.

H. Bourlard, Y. Konig, and N. Morgan. REMAP: recursive estimation and maximization of a posteriori probabilities in connectionist speech recognition. In *Proceedings of the European Conference on Speech Communication and Technology (EUROSPEECH)*, 1995.

S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

J. S Bridle. Alpha-nets: a recurrent 'neural'network architecture with a hidden Markov model interpretation. *Speech Communication*, 9(1):83–92, 1990a.

J.S. Bridle. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In *Neuro-computing: Algorithms, Architectures and Applications*, pages 227–236. 1990b.

C. Bucilua, R. Caruana, and A. Niculescu-Mizil. Model compression. In *Proceedings of the 12th International conference on Knowledge discovery and data mining (SIGKDD)*, pages 535–541. ACM, 2006.

D. Can and M. Saraclar. Lattice Indexing for Spoken Term Detection. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(8):2338–2347, Nov 2011.

R. Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals. Listen, attend and spell. *arXiv preprint arXiv:1508.01211*, 2015.

S. Chang and N. Morgan. Robust CNN-based speech recognition with Gabor filter kernels. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 905–909, 2014.

J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio. Attention-based models for speech recognition. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, pages 577–585, 2015.

R. Collobert and J. Weston. A unified architecture for natural language processing: deep neural networks with multitask learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 160–167, 2008.

R. Collobert, K. Kavukcuoglu, and C. Farabet. Torch7: A Matlab-like Environment for Machine Learning. In *BigLearn, NIPS Workshop*, 2011a.

R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537, 2011b.

M. Cooke, P. Green, L. Josifovski, and A. Vizinho. Robust automatic speech recognition with missing and unreliable acoustic data. *Speech communication*, 34(3):267–285, 2001.

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

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

G. E. Dahl, T. N. Sainath, and G. E. Hinton. Improving deep neural networks for LVCSR using rectified linear units and dropout. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8609–8613. IEEE, 2013.

S. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 28(4):357–366, 1980.

L. Deng and D. O'Shaughnessy. *Speech processing: a dynamic and optimization-oriented approach*. CRC Press, 2003.

L. Deng and J. C. Platt. Ensemble deep learning for speech recognition. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 1915–1919, 2014.

L. Deng, O. Abdel-Hamid, and D. Yu. A deep convolutional neural network using heterogeneous pooling for trading acoustic invariance with phonetic confusion. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6669–6673. IEEE, 2013.

J. L. Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

Y. Ephraim and William J J Roberts. Revisiting autoregressive hidden markov modeling of speech signals. *IEEE Signal Processing Letters*, 12(2):166–169, February 2005.

T. Fawcett. An introduction to ROC analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

J. G Fiscus, J. Ajot, J.S. Garofolo, and G. Doddingtion. Results of the 2006 spoken term detection evaluation. In *Proceedings of SIGIR*, volume 7, pages 51–57. Citeseer, 2007.

J. Flores and S. J. Young. Continuous speech recognition in noise using spectral subtraction and HMM adaptation. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, pages I–409. IEEE, 1994.

G. D. Forney. The Viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

E.L. Fosler and J. Morris. Crandem systems: Conditional random field acoustic models for hidden Markov models. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4049 –4052, April 2008.

S. Furui. Cepstral analysis technique for automatic speaker verification. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 29(2):254–272, 1981.

S. Furui. Speaker-independent isolated word recognition based on emphasized spectral dynamics. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 11, pages 1991–1994. IEEE, 1986.

S. Furui. Toward robust speech recognition under adverse conditions. In *Speech Processing in Adverse Conditions*, 1992.

M. Gales and P. C. Woodland. Mean and variance adaptation within the MLLR framework. *Computer Speech & Language*, 10(4):249–264, 1996.

M. Gales and S. Young. Robust continuous speech recognition using parallel model combination. *IEEE Transactions on Speech and Audio Processing*, 4(5):352–359, 1996.

M. Gales and S. Young. The Application of Hidden Markov Models in Speech Recognition. *Found. Trends Signal Process.*, 1(3):195–304, January 2007.

P. N. Garner. SNR Features for Automatic Speech Recognition. In *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 182–187, November 2009.

J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett. DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1. *NASA STI/Recon Technical Report N*, 93, 1993.

J. R. Glass. A probabilistic framework for segment-based speech recognition. *Computer Speech & Language*, 17(2):137–152, 2003.

X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 249–256, 2010.

B. Gold, N. Morgan, and D. Ellis. *Speech and audio signal processing: processing and perception of speech and music.* John Wiley & Sons, 2011.

P. Golik, Z. Tüske, R. Schlüter, and H. Ney. Convolutional Neural Networks for Acoustic Modeling of Raw Time Signal in LVCSR. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 26–30, 2015.

A. Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

A. Graves and N. Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1764–1772, 2014.

A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5):602–610, 2005.

A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 369–376. ACM, 2006.

A. Graves, A. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6645–6649. IEEE, 2013.

K. Greer, B. Lowerre, and L. Wilcox. Acoustic pattern matching and beam searching. In *Proceeding of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 7, pages 1251–1254. IEEE, 1982.

W. Guangsen and K. C. Sim. Sequential classification criteria for NNs in automatic speech recognition. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 441–444, 2011.

P. Haffner. Connectionist word-level classification in speech recognition. In *Proceeding of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, pages 621–624, 1992.

A. Hagen. Robust speech recognition based on multi-stream processing. Technical report, 2001.

K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. *arXiv preprint arXiv:1512.03385*, 2015.

H. Hermansky. Perceptual linear predictive (PLP) analysis of speech. *The Journal of the Acoustical Society of America*, 87:1738, 1990.

H. Hermansky. Should recognizers have ears? *Speech communication*, 25(1):3–27, 1998.

H. Hermansky and P. Fousek. Multi-resolution RASTA filtering for TANDEM-based ASR. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2005.

Y. Hifny and S. Renals. Speech recognition using augmented conditional random fields. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(2):354–365, 2009.

G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, and T. N. Sainath. Deep neural networks for acoustic modeling in speech recognition: the Shared Views of Four Research Groups. *Signal Processing Magazine, IEEE*, 29(6):82–97, 2012.

G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

G. E. Hinton and K.J. Lang. The development of the time-delay neural network architecture for speech recognition. *Techn. Rep. Carnegie Mellon Univ. CMU-CS*, pages 88–152, 1988.

G. E. Hinton, S. Osindero, and Y. W. Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

H.-G. Hirsch. Experimental framework for the performance evaluation of speech recognition front-ends on a large vocabulary task. *ETSI STQ Aurora DSR Working Group*, 2002a.

H.-G. Hirsch. The influence of speech coding on recognition performance in telecommunication networks. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, 2002b.

H.-G. Hirsch and D. Pearce. The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions. In *ASR2000-Automatic Speech Recognition: Challenges for the new Millenium ISCA Tutorial and Research Workshop (ITRW)*, 2000.

H.-G. Hirsch and D. Pearce. Applying the advanced ETSI frontend to the aurora-2 task. Technical report, 2006. version 1.1.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

F. Hönig, G. Stemmer, C. Hacker, and F. Brugnara. Revising perceptual linear prediction (plp). In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 2997–3000, 2005.

K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Y. Hoshen, R. J. Weiss, and K. W. Wilson. Speech acoustic modeling from raw multichannel waveforms. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4624–4628. IEEE, 2015.

S. Ikbal. *Nonlinear feature transformations for noise robust speech recognition*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, 2004.

D. Imseng, H. Bourlard, H. Caesar, P. N. Garner, G. Lecorvé, A. Nanchen, and others. MediaParl: Bilingual mixed language accented speech database. In *Proceedings of the IEEE Spoken Language Technology Workshop (SLT)*, pages 263–268, 2012.

S. Ioffe and C. Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *CoRR*, abs/1502.03167, 2015. URL http://arxiv.org/abs/1502.03167.

N. Jaitly and G. Hinton. Learning a better representation of speech soundwaves using restricted Boltzmann machines. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5884–5887, 2011.

F. Jelinek. *Statistical methods for speech recognition*. MIT press, 1997.

B.-H. Juang and L. R. Rabiner. The segmental K-means algorithm for estimating parameters of hidden Markov models. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 38(9): 1639–1641, 1990.

S. M. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions onAcoustics, Speech and Signal Processing*, 35(3): 400–401, 1987.

J. Keshet, D. Chazan, and B. Bobrovsky. Plosive spotting with margin classifiers. In *Proceedings of the European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 1637–1640, 2001.

J. Keshet, D. Grangier, and S. Bengio. Discriminative keyword spotting. *Speech Communication*, 51(4):317–329, 2009.

B. Kingsbury. Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3761–3764. IEEE, 2009.

R. Kneser and H. Ney. Improved backing-off for m-gram language modeling. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, pages 181–184. IEEE, 1995.

A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet classification with deep convolutional neural networks. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, pages 1106–1114, 2012.

Y. Kubo, T. Hori, and A. Nakamura. Integrating Deep Neural Networks into Structural Classification Approach based on Weighted Finite-State Transducers. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2012.

J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2001.

H. Larochelle, Y. Bengio, J. Louradour, and P. Lamblin. Exploring strategies for training deep neural networks. *The Journal of Machine Learning Research*, 10:1–40, 2009.

G. Lathoud, M. Magimai-Doss, B. Mesot, and H. Bourlard. Unsupervised spectral subtraction for noise-robust ASR. In *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 343–348. IEEE, 2005.

G. Lecorvé and P. Motlicek. Conversion of Recurrent Neural Network Language Models to Weighted Finite State Transducers for Automatic Speech Recognition. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 131–134, 2012.

Y. LeCun. Generalization and Network Design Strategies. In R. Pfeifer, Z. Schreter, F. Fogelman, and L. Steels, editors, *Connectionism in Perspective*, Zurich, Switzerland, 1989. Elsevier.

Y. LeCun, J. S. Denker, S. A. Solla, R. E. Howard, and L. D. Jackel. Optimal brain damage. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, volume 89, 1989.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

H. Lee, P. Pham, Y. Largman, and A. Y. Ng. Unsupervised feature learning for audio classification using convolutional deep belief networks. In *Advances in Neural Information Processing Systems 22*, pages 1096–1104, 2009.

K. F. Lee and H. W. Hon. Speaker-independent phone recognition using hidden Markov models. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37(11):1641–1648, 1989.

B. Li and K. C. Sim. Noise adaptive front-end normalization based on vector Taylor series for deep neural networks in robust speech recognition. In *Proceedings of the IEEE International*

Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 7408–7412. IEEE, 2013.

J. Li, L. Deng, D. Yu, Y. Gong, and A. Acero. High-performance HMM adaptation with joint compensation of additive and convolutive distortions via vector Taylor series. In *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 65–70. IEEE, 2007.

P. Lockwood and J. Boudy. Experiments with a nonlinear spectral subtractor (NSS), hidden Markov models and the projection, for robust speech recognition in cars. *Speech Communication*, 11(2-3):215–228, 1992.

L. Lu, L. Kong, C. Dyer, N. A. Smith, and S. Renals. Segmental Recurrent Neural Networks for End-to-end Speech Recognition. *arXiv preprint arXiv:1603.00223*, 2016.

A. L. Maas, Q. V. Le, T. M. O'Neil, O. Vinyals, P. Nguyen, and A. Y. Ng. Recurrent Neural Networks for Noise Reduction in Robust ASR. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 22–25, 2012.

W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

B. Mesot and D. Barber. Switching Linear Dynamical Systems for Noise Robust Speech Recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(6):1850–1858, August 2008.

T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur. Recurrent neural network based language model. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, volume 2, page 3, 2010.

H. Misra, J. Vepa, and H. Bourlard. Multi-stream ASR: an oracle perspective. In *Proceedings of ISCA International Conference on Spoken Language Processing (ICSLP)*, 2006.

V. Mitra, W. Wang, H. Franco, Y. Lei, C. Bartels, and M. Graciarena. Evaluating robust features on deep neural networks for speech recognition in noisy and channel mismatched conditions. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 895–899, 2014.

V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu. Recurrent models of visual attention. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, pages 2204–2212, 2014.

A. Mohamed, G. Dahl, and G. Hinton. Deep belief networks for phone recognition. In *NIPS Workshop on Deep Learning for Speech Recognition and Related Applications*, 2009.

A. Mohamed, D. Yu, and L. Deng. Investigation of full-sequence training of deep belief networks for speech recognition. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, volume 10, pages 2846–2849, 2010.

A. Mohamed, T. N. Sainath, G. Dahl, B. Ramabhadran, G. E. Hinton, and M. A. Picheny. Deep belief networks using discriminative features for phone recognition. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 5060–5063. IEEE, 2011.

A. Mohamed, G.E. Dahl, and G. Hinton. Acoustic Modeling Using Deep Belief Networks. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):14 –22, jan. 2012.

N. Morgan and H. Bourlard. Generalization and parameter estimation in feedforward nets: Some experiments. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, pages 630–637, 1989.

N. Morgan and H. Bourlard. Continuous speech recognition. *Signal Processing Magazine, IEEE*, 12(3):24 –42, May 1995.

J. Morris and E. Fosler-Lussier. Conditional Random Fields for Integrating Local Discriminative Classifiers. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(3):617–628, March 2008.

V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 807–814, 2010.

J. J. Odell. *The Use of Context in Large Vocabulary Speech Recognition*. PhD thesis, University of Cambridge, 1995.

D. O'Shaughnessy. *Speech communication: human and machine*. Universities press, 1987.

D. Palaz, R. Collobert, and M. Magimai.-Doss. Estimating Phoneme Class Conditional Probabilities from Raw Speech Signal using Convolutional Neural Networks. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 1766–1770, September 2013a.

D. Palaz, R. Collobert, and M. Magimai.-Doss. End-to-end Phoneme Sequence Recognition using Convolutional Neural Networks. *NIPS Deep Learning Workshop*, December 2013b.

D. Palaz, M. Magimai Doss, and R. Collobert. Learning linearly separable features for speech recognition using convolutional neural networks. *ICLR Workshop*, April 2014a.

D. Palaz, M. Magimai-Doss, and R. Collobert. Joint phoneme segmentation inference and classification using CRFs. In *Proceedings of GlobalSIP*, pages 587–591. IEEE, 2014b.

D. Palaz, M. Magimai-Doss, and R. Collobert. Analysis of CNN-based Speech Recognition System using Raw Speech as Input. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 11–15, 2015a.

D. Palaz, M. Magimai.-Doss, and R. Collobert. Convolutional Neural Networks-based Continuous Speech Recognition using Raw Speech Signal. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4295–4299, April 2015b.

D. Palaz, G. Synnaeve, and R. Collobert. Jointly Learning to Locate and Classify Words using Convolutional Networks. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH), preprint*, 2016.

V. Panayotov, G. Chen, D. Povey, and S. Khudanpur. Librispeech: an ASR corpus based on public domain audio books. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015.

P. O. Pinheiro and R. Collobert. From Image-level to Pixel-level Labeling with Convolutional Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1713–1721, 2015.

A. Poritz. Linear predictive hidden Markov models and the speech signal. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 7, pages 1291–1294, May 1982.

D. Povey and P. C. Woodland. Minimum phone error and I-smoothing for improved discriminative training. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, pages I–105. IEEE, 2002.

D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely. The Kaldi Speech Recognition Toolkit. In *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, December 2011.

R. Prabhavalkar and E. Fosler-Lussier. Backpropagation training for multilayer conditional random field based phone recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5534–5537. IEEE, 2010.

L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257 –286, February 1989.

L. R. Rabiner and B.-H. Juang. *Fundamentals of Speech Recognition*. Prentice-Hall, Inc., 1993.

L. R. Rabiner and R. W. Schafer. *Digital processing of speech signals*. Prentice Hall, 1978.

A. Ragni and M. J.-F. Gales. Inference algorithms for generative score-spaces. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4149–4152. IEEE, 2012.

B. Raj, M. L. Seltzer, and R. M. Stern. Robust speech recognition: the case for restoring missing features. In *Proc. of Eurospeech, The Workshop on Consistent and Reliable Acoustic Cues, Aalborg, Denmark*, 2001.

S. P. Rath, D. Povey, K. Veselỳ, and J. Cernockỳ. Improved feature processing for deep neural networks. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 109–113, 2013.

M. Razavi and M. Magimai.-Doss. On Recognition of Non-Native Speech Using Probabilistic Lexical Model. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2014.

M. Razavi, R. Rasipuram, and M. Magimai.-Doss. On Modeling Context-Dependent Clustered States: Comparing HMM/GMM, Hybrid HMM/ANN and KL-HMM Approaches. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.

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(1):161–174, 1994.

M. D. Richard and R. P. Lippmann. Neural network classifiers estimate Bayesian a posteriori probabilities. *Neural computation*, 3(4):461–483, 1991.

T. Robinson. An Application of Recurrent Nets to Phone Probability Estimation. *IEEE Transactions on Neural Networks*, 5:298–305, 1994.

J.R. Rohlicek, W. Russell, S. Roukos, and H. Gish. Continuous hidden Markov modeling for speaker-independent word spotting. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 627 –630 vol.1, May 1989.

R.C. Rose and D.B. Paul. A hidden Markov model based keyword recognition system. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 129 –132 vol.1, April 1990. doi: 10.1109/ICASSP.1990.115555.

F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. Technical report, DTIC Document, 1985.

T. N. Sainath, B. Kingsbury, and B. Ramabhadran. Auto-encoder bottleneck features using deep belief networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4153–4156. IEEE, 2012.

T. N. Sainath, B. Kingsbury, A. Mohamed, G. E. Dahl, G. Saon, H. Soltau, T. Beran, A. Y. Aravkin, and B. Ramabhadran. Improvements to deep convolutional neural networks for LVCSR. In *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 315–320. IEEE, 2013a.

T. N. Sainath, A. Mohamed, B. Kingsbury, and B. Ramabhadran. Deep convolutional neural networks for LVCSR. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8614–8618, 2013b.

T. N. Sainath, O. Vinyals, A. Senior, and H. Sak. Convolutional, long short-term memory, fully connected deep neural networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4580–4584. IEEE, 2015a.

T. N. Sainath, R. J. Weiss, A. Senior, K. W. Wilson, and O. Vinyals. Learning the Speech Front-end With Raw Waveform CLDNNs. *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2015b.

T.N. Sainath, B. Kingsbury, A.-R. Mohamed, and B. Ramabhadran. Learning filter banks within a deep neural network framework. In *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 297–302, Dec 2013c. doi: 10.1109/ASRU.2013.6707746.

M. Saraclar and R. Sproat. Lattice-based search for spoken utterance retrieval. *Urbana*, 51: 61801, 2004.

M. R. Schroeder and B. S. Atal. Code-excited linear prediction (CELP): High-quality speech at very low bit rates. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'85.*, volume 10, pages 937–940. IEEE, 1985.

M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

R. Schwartz, Y. L. Chow, O. Kimball, S. Roucos, M. Krasner, and J. Makhoul. Context-dependent modeling for acoustic-phonetic recognition of continuous speech. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 10, pages 1205–1208. IEEE, 1985.

F. Seide, G. Li, and D. Yu. Conversational speech transcription using context-dependent deep neural networks. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 437–440, 2011.

M. L. Seltzer, D. Yu, and Y. Wang. An investigation of deep neural networks for noise robust speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7398–7402. IEEE, 2013.

S. Sharma, D. Ellis, S. Kajarekar, P. Jain, and H. Hermansky. Feature extraction using non-linear transformation for robust speech recognition on the aurora database. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 2, pages II1117–II1120. IEEE, 2000.

H. Sheikhzadeh and L. Deng. Waveform-based speech recognition using hidden filter models: Parameter selection and sensitivity to power normalization. *IEEE Transactions on Speech and Audio Processing*, 2(1):80–89, 1994.

D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587): 484–489, January 2016.

S. Sivadas, Z. Wu, and M. Bin. Investigation of Parametric Rectified Linear Units for Noise Robust Speech Recognition. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2015.

N. Srivastava. Improving neural networks with dropout. Master's thesis, University of Toronto, 2013.

P. Swietojanski, A. Ghoshal, and S. Renals. Convolutional Neural Networks for Distant Speech Recognition. *IEEE Signal Processing Letters*, 21(9):1120–1124, September 2014.

I. Szöke, P. Schwarz, P. Matějka, and M. Karafiát. Comparison of keyword spotting approaches for informal continuous speech. In *Proceedings of the European Conference on Speech Communication and Technology (EUROSPEECH)*, 2005.

S. Tamura and A. Waibel. Noise reduction using connectionist models. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 553–556. IEEE, 1988.

G. Trigeorgis, F. Ringeval, R. Brueckner, E. Marchi, M. A. Nicolaou, B. Schuller, and S. Zafeiriou. Adieu features? End-To-End Speech Emotion Recognition using a Deep Convolutional Recurrent Network. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

Z. Tüske, P. Golik, R. Schlüter, and H. Ney. Acoustic Modeling with Deep Neural Networks Using Raw Time Signal for LVCSR. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 890–894, Singapore, September 2014.

G. Urban, K. J. Geras, S. E. Kahou, O. Aslan, S. Wang, R. Caruana, A. Mohamed, M. Philipose, and M. Richardson. Do Deep Convolutional Nets Really Need to be Deep (Or Even Convolutional)? *arXiv preprint arXiv:1603.05691*, 2016.

A Varga and R Moore. Hidden Markov model decomposition of speech and noise. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 845–848. IEEE, 1990.

K. Vesely, A. Ghoshal, L. Burget, and D. Povey. Sequence-discriminative training of deep neural networks. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 2345–2349, 2013.

O. Viikki and K. Laurila. Cepstral domain segmental feature vector normalization for noise robust speech recognition. *Speech Communication*, 25(1):133–147, 1998.

P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1096–1103. ACM, 2008.

O. Vinyals and S. V. Ravuri. Comparing multilayer perceptron to deep belief network tandem features for robust ASR. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 4596–4599. IEEE, 2011.

A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K.J. Lang. Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37(3):328 –339, mar 1989.

C. Weng, D. Yu, S. Watanabe, and B. F. Juang. Recurrent deep neural networks for robust speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5532–5536. IEEE, 2014.

G. Williams and S. Renals. Confidence measures from local posterior probability estimates. *Computer Speech & Language*, 13(4):395–411, 1999.

J.G. Wilpon, L.R. Rabiner, C.-H. Lee, and E.R. Goldman. Automatic recognition of keywords in unconstrained speech using hidden Markov models. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 38(11):1870 –1878, November 1990.

P.C. Woodland, J.J. Odell, V. Valtchev, and S.J. Young. Large vocabulary continuous speech recognition using HTK. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume ii, pages 125–128, apr 1994.

S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, V. Valtchev, and P. Woodland. The HTK book. *Cambridge University Engineering Department*, 3, 2002.

S. J. Young, J. J. Odell, and P. C. Woodland. Tree-based state tying for high accuracy acoustic modelling. In *Proceedings of the workshop on Human Language Technology*, pages 307–312. Association for Computational Linguistics, 1994.

J. Yousafzai, Z. Cvetkovic, and P. Sollich. Tuning support vector machines for robust phoneme classification with acoustic waveforms. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 2391–2394, 2009.

D. Yu and M. L. Seltzer. Improved Bottleneck Features Using Pretrained Deep Neural Networks. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, volume 237, page 240, 2011.

P. Yu and F. Seide. A hybrid-word/phoneme-based approach for improved vocabulary-independent search in spontaneous speech. In *Proceedings of the Annual Conference* *of the International Speech Communication Association (INTERSPEECH)*, pages 293–296, 2004.

X. Zhang, J. Trmal, D. Povey, and S. Khudanpur. Improving deep neural network acoustic models using generalized maxout networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 215–219. IEEE, 2014.

G. Zweig and P. Nguyen. A segmental CRF approach to large vocabulary continuous speech recognition. In *Proceedings of the IEEE Workshop on Automatic Speech Recognition & Understanding, (ASRU).*, pages 152–157. IEEE, 2009.

Dimitri PALAZ

Ph.D. in Electrical Engineering

Rue de la Maladière 8
1920 Martigny, Switzerland
📱 *+41 79 825 56 20*
☎ *+41 27 721 77 03*
✉ *dimitri.palaz@idiap.ch*
🖱 *people.idiap.ch/dpalaz*

Education

2011–2016 **Ph.D. in Electrical Engineering**, *École Polytechnique Fédérale de Lausanne (EPFL)*, Lausanne, Switzerland.

- Topic: Towards End-to-End Speech Recognition
- Supervisors: Prof. Hervé Bourlard, Dr. Ronan Collobert and Dr. Mathew Magimai.-Doss.

2005–2011 **M.Sc. in Electrical and Electronics Engineering**, *École Polytechnique Fédérale de Lausanne (EPFL)*, Lausanne, Switzerland.

- Topic: Dictionary learning for sparse stereo image representation and encoding.
- Supervisor: Prof. Pascal Frossard.

2009–2010 **Minor in Space Technology**, *École Polytechnique Fédérale de Lausanne (EPFL)*, Lausanne, Switzerland.

- Topic: Change detection analysis using multi-temporal SAR remote sensing data.
- Supervisor: Dr. Maurice Borgeaud.

Experience

2015 **Research Intern**, *Facebook Artificial Intelligence Research*, Menlo Park, CA, U.S.A..

- Supervised by Dr. Ronan Collobert and Dr. Gabriel Synnaeve.

2011–2016 **Research Assistant**, *Idiap Research Institute*, Martigny, Switzerland.

2007–2010 **Teaching Assistant**, *École Polytechnique Fédérale de Lausanne (EPFL)*, Lausanne, Switzerland.

- Remote sensing of the earth by satellite, Introduction to C++ programmation, Electronics.

Professional Activities

- Reviewer for ICML 2016
- Student projects supervision
- IEEE and IEEE SPS Member

Languages

French	Native Language
English	Fluent
German	Conversational

Computer skills

Programming	Lua, C/C++, Perl, Bash, CUDA, Python
Science tools	Torch7, MatLab, HTK, Kaldi
Office	Latex, Beamer, LibreOfffice.

Publications

Peer-reviewed Conferences

Dimitri Palaz, Gabriel Synnaeve and Ronan Collobert, “Jointly Learning to Locate and Classify Words using Convolutional Networks”, *17th Annual Conference of the International Speech Communication Association*, 2016, preprint

Dimitri Palaz, Mathew Magimai-Doss and Ronan Collobert, “Analysis of CNN-based Speech Recognition System using Raw Speech as Input”, in *16th Annual Conference of the International Speech Communication Association*, 2015

Dimitri Palaz, Mathew Magimai-Doss and Ronan Collobert, “Convolutional neural networks-based continuous speech recognition using raw speech signal”, in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015

Dimitri Palaz, Mathew Magimai-Doss and Ronan Collobert, “Joint phoneme segmentation inference and classification using CRFs”, in *Global Conference on Signal and Information Processing (GlobalSIP)*, 2014.

Dimitri Palaz, Ronan Collobert and Mathew Magimai-Doss, “Estimating phoneme class conditional probabilities from raw speech signal using convolutional neural networks”, in *14th Annual Conference of the International Speech Communication Association*, pp. 1766–1770, 2013.

Dimitri Palaz, Ivana Tosic and Pascal Frossard, “Sparse stereo image coding with learned dictionaries”, in *International Conference on Image Processing (ICIP)*, pp. 133-136, 2011.

Peer-reviewed Workshops

Dimitri Palaz, Mathew Magimai.-Doss, and Ronan Collobert, “Learning linearly separable features for speech recognition using convolutional neural networks”, in *ICLR workshop*, 2015.

Dimitri Palaz, Ronan Collobert and Mathew Magimai.-Doss, “End-to-end phoneme sequence recognition using convolutional neural networks”, in *NIPS deep learning workshop*, 2013.

Symposiums

Maurice Borgeaud, Pierre Deleglise and **Dimitri Palaz**, “Monitoring of land cover change using SAR and optical data from the ESA Rolling Archives”, in *ESA Living Planet Symposium*, 2010.

Manuscript in preparation

Dimitri Palaz, Mathew Magimai-Doss, and Ronan Collobert, “End-to-End Acoustic Modeling using Convolutional Neural Networks for Automatic Speech Recognition”, Manuscript in preparation.

Dimitri Palaz, Ronan Collobert and Mathew Magimai-Doss, “Robust Raw Speech-based ASR using Normalized Convolutional Neural Networks”, Manuscript in preparation.