



UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

DEPARTAMENTO DE TECNOLOGÍA Y DE LAS COMUNICACIONES



# AUTOMATIC LANGUAGE RECOGNITION USING DEEP NEURAL NETWORKS

*–TRABAJO FIN DE MÁSTER–*

*RECONOCIMIENTO AUTOMÁTICO DE IDIOMA MEDIANTE REDES  
NEURONALES PROFUNDAS*

**Author: Alicia Lozano Díez**

**Ingeniera en Informática y Licenciada en Matemáticas,  
Universidad Autónoma de Madrid**

A Thesis submitted for the degree of:

*Máster Universitario en Investigación e Innovación en TIC  
(Master of Science)*

Madrid, September 2013

## Colophon

This book was typeset by the author using L<sup>A</sup>T<sub>E</sub>X2<sub>ε</sub>. The main body of the text was set using a 11-points Computer Modern Roman font. All graphics and images were included formatted as Encapsulated Postscript (<sup>TM</sup> Adobe Systems Incorporated). The final postscript output was converted to Portable Document Format (PDF) and printed.

Copyright © 2013 by Alicia Lozano Díez. All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, recording, or any information storage and retrieval system, without permission in writing from the author. Universidad Autonoma de Madrid has several rights in order to reproduce and distribute electronically this document.

Departamento: Tecnología Electrónica y de las Comunicaciones  
Escuela Politécnica Superior  
Universidad Autónoma de Madrid (UAM), SPAIN

Título: Automatic Language Recognition Using Deep Neural Networks

Autor: **Alicia Lozano Díez**  
Ingeniera en Informática y Licenciada en Matemáticas  
(Universidad Autónoma de Madrid)

Director: **Dr. Javier González Domínguez**  
Doctor Ingeniero de Telecomunicación  
Universidad Autónoma de Madrid, SPAIN

Tutor: **Prof. Joaquín González Rodríguez**  
Doctor Ingeniero de Telecomunicación  
(Universidad Politécnica de Madrid)  
Universidad Autónoma de Madrid, SPAIN

Fecha: 23 de Septiembre de 2013

Tribunal: **Prof. Joaquín González Rodríguez**  
Universidad Autónoma de Madrid, SPAIN

**Dr. Daniel Ramos Castro**  
Universidad Autónoma de Madrid, SPAIN

**Prof. Doroteo Torre Toledano**  
Universidad Autónoma de Madrid, SPAIN

Calificación:



---

The research described in this Master Thesis was carried out within the Biometric Recognition Group – ATVS at the Dept. of Tecnología Electrónica y de las Comunicaciones, Escuela Politécnica Superior, Universidad Autónoma de Madrid. The project was partially funded by a PhD scholarship from Servicio de Investigación (Universidad Autónoma de Madrid).



# Abstract

IN RECENT YEARS, deep learning has been arisen as a new paradigm within machine learning field. In particular, Deep Neural Networks (DNNs) are an important part of this new paradigm. This set of architectures has properties that make them suitable for difficult tasks among which it can be highlighted automatic language recognition (or Spoken Language Recognition, SLR). Their capability to model complex functions in high-dimensional spaces and to get a good representation of the input data makes these architectures and algorithms proper for processing complex signals as, for instance, the voice signal. Thereby, they can be used as a technique to provide an automatic way to distinguish the language that has been used in a specific segment of speech.

This Master Thesis is intended to provide a new approach that, combining both deep learning and automatic language recognition fields, improves the SLR task by getting a better representation of voice signals for classification purposes so that it can be identified which language has been used in that voice signal.

In order to do this, both DNNs and state-of-art SLR systems have been studied thoroughly. Firstly, it has been reviewed the application of DNNs to speech recognition tasks. Then, convolutional deep neural networks, in particular, have been adapted to the language recognition problem and their performance has been evaluated on a challenging dataset such as NIST LRE 2009 (National Institute of Standards and Technology Language Recognition Evaluation).

Although some results do not always outperform the reference system that has been considered in the experimental part of this work, the new approach based on DNNs can be seen as a *starting point* to improve current SLR systems.



A MIS PADRES.

A MI HERMANA.

A MIS ABUELAS.

A “LOS DE SIEMPRE”.





# Acknowledgements

THIS MASTER THESIS summarizes the work carried out during the last year during my Master studies with the Biometric Recognition Group - ATVS at the Dept. of Tecnología Electrónica y de las Comunicaciones (Escuela Politécnica Superior, Universidad Autónoma de Madrid).

Firstly, I would like to thank my advisors Dr. Javier González Domínguez and Prof. Joaquín González Rodríguez for their guidance and support since I started working in the group. They gave me the opportunity of working in this research line and have trusted me since I started. This has made my motivation grows during this year working with them.

In the framework of the ATVS research group, I have received also the support from Prof. Javier Ortega García and from Prof. Doroteo Torre Toledano, who have given me good advises and let me learn from their experience.

Of course, I would like to thank my workmates at ATVS: Álvaro, Javier (Franco), Daniel, Ester, Rubén (Vera), Marta, Pedro, Ruifang, Fernando, Rubén (Zazo) and Julián. They make my daily work be easier and funnier.

I have to thank also the people that have shared with me lots of moments before this year. I want to say thanks to my lifelong friends (Javi, Gabri, Marta, Gonzalo, Vito, Jaime, Alba, Adalberto, Carla, Pablo, Bea...) who spend with me most of my (and theirs) free time, sharing the best moments of our lives; and my friends from university (Pilar, Álvaro, Mambri, Alba...) who passed with me through those hard five years.

But I cannot finish without thanking my parents (Jose y Grani) and my sister (Elena), of course. They have given me everything and have made me be who I am... What else can I ask for? Thank you very much!

*Alicia Lozano Díez*

*Madrid, September 2013*



# Contents

<b>Abstract</b>	<b>v</b>
<b>Acknowledgements</b>	<b>ix</b>
<b>List of Figures</b>	<b>xiii</b>
<b>List of Tables</b>	<b>xiv</b>
<b>1. Introduction</b>	<b>1</b>
1.1. Language Recognition . . . . .	1
1.2. Deep Neural Networks . . . . .	3
1.3. Motivation . . . . .	4
1.4. Goals of the Master Thesis . . . . .	4
1.5. Outline of the Dissertation . . . . .	5
<b>2. Related Works and State of the Art</b>	<b>7</b>
2.1. Automatic Language Recognition Systems . . . . .	7
2.1.1. Why Language Recognition? . . . . .	7
2.1.2. Spectral Systems: Acoustic Approaches . . . . .	8
2.1.3. High Level Systems: Phonotactic Approaches . . . . .	9
2.2. Deep Neural Networks . . . . .	11
2.2.1. A New Machine Learning Paradigm . . . . .	12
2.2.2. Supervised and Unsupervised Learning . . . . .	13
2.2.3. From Shallow to Deep Architectures . . . . .	14
2.2.4. Training Algorithms for Deep Architectures . . . . .	17
2.2.5. Case of Success: Deep Convolutional Networks . . . . .	18
<b>3. Deep Neural Networks Applied to Language Recognition</b>	<b>21</b>
3.1. DNNs Applied to Speech Recognition . . . . .	22
3.2. Proposed Method: DNNs Applied to Language Recognition . . . . .	25

<b>4. Experiments and Results</b>	<b>29</b>
4.1. Reference System . . . . .	29
4.2. Database Description . . . . .	30
4.3. Experimental Framework . . . . .	31
4.4. Results . . . . .	33
4.4.1. “One vs. One” Experiments ( <i>Language Pairs</i> ) . . . . .	33
4.4.2. “All vs. All” Experiments ( <i>Closed-set</i> ) . . . . .	37
4.4.3. “One vs. All” Experiments . . . . .	42
<b>5. Conclusions and Future Work</b>	<b>43</b>
5.1. Conclusions . . . . .	43
5.2. Future Work . . . . .	45
<b>A. Short Biography</b>	<b>47</b>

# List of Figures

1.1.	General structure of a language recognition system. . . . .	3
2.1.	Computation of the SDC feature vector at frame $t$ for parameters $N$ - $d$ - $P$ - $k$ [extracted from Torres-carrasquillo <i>et al.</i> [2002]]. . . . .	9
2.2.	Verification task in a PRLM system [extracted from Gonzalez-Dominguez <i>et al.</i> [2010]]. . . . .	11
2.3.	Expression computed by a deep architecture with sums and products [extracted from Bengio [2009]]. . . . .	15
2.4.	Same expression from figure 2.3 computed by a shallow architecture with sums and products. . . . .	16
2.5.	Example of deep architecture for classification. . . . .	17
2.6.	Training algorithm for a deep belief network [extracted from Hinton <i>et al.</i> [2012a]].	18
3.1.	The 35 misclassified digits [extracted from Ciresan <i>et al.</i> [2010]]. . . . .	22
3.2.	Examples of object classification predictions [extracted from Krizhevsky <i>et al.</i> [2012]]. . . . .	23
3.3.	Spectrograms of the same phoneme for female and male speakers and their representations obtained by a convolutional DBN [extracted from Lee <i>et al.</i> [2009b]].	24
3.4.	Example of a convolutional DNN structure used in the experimental part. . . . .	26
4.1.	DET curves corresponding to results of the Bosnian vs Croatian experiment. . .	35
4.2.	DET curves corresponding to pairs results of the reference system with 1024 Gaussian components (top) and the best DNN models according to the EER for each language pair (down). . . . .	36



# List of Tables

4.1. Data used to train the elements of the reference system [extracted from Gonzalez-Dominguez <i>et al.</i> [2010]]. . . . .	30
4.2. Target languages in the NIST LRE09 evaluation. . . . .	30
4.3. Amount of available data for each language used for development in the experiments performed. Each file corresponds with approximately 150 seconds of speech. . . . .	32
4.4. Configuration parameters for the experiments performed on the language pair Hindi - Ukrainian. . . . .	33
4.5. Results experiment Hindi vs. Ukrainian. . . . .	34
4.6. Results experiment Turkish vs. Vietnamese. . . . .	34
4.7. Results experiment Bosnian vs. Croatian. . . . .	35
4.8. Configuration parameters for the performed experiments based on SDC representation. . . . .	38
4.9. Results “all vs. all” experiments based on SDC representation (ABCCHU experiments). . . . .	38
4.10. Confusion matrix for the reference system (512 Gaussian components). Languages Amharic, Bosnian, Cantonese, Croatian, Hindi and Urdu. . . . .	38
4.11. Confusion matrix for the reference system (1024 Gaussian components). Languages Amharic, Bosnian, Cantonese, Croatian, Hindi and Urdu. . . . .	38
4.12. Confusion matrix for the experiment <i>ABCCHU-EVAL-TEST</i> (test evaluation data). . . . .	39
4.13. Confusion matrix for the experiment <i>ABCCHU-DEV-TEST</i> (test part of the development dataset). . . . .	39
4.14. Results “all vs. all” experiments based on MFB representation (CHHTUV experiments). . . . .	40
4.15. Confusion matrix for the reference system (512 Gaussian components). Languages: Creole, Hausa, Hindi, Turkish, Ukrainian and Vietnamese. . . . .	41
4.16. Confusion matrix for the reference system (1024 Gaussian components). Languages: Creole, Hausa, Hindi, Turkish, Ukrainian and Vietnamese. . . . .	41
4.17. Confusion matrix for the experiment <i>CHHTUV-300</i> , model of 300 frames. . . . .	41

4.18. Performance of reference system (1024 Gaussian components) ( <i>ATVS3</i> , Gonzalez-Dominguez [2011]) and “one vs. all” models using convolutional DNNs on development dataset (per language). . . . .	42
---	----



# Chapter 1

## Introduction

AUTOMATIC SPEECH PROCESSING AND MACHINE LEARNING are two research fields strongly linked due to their complementarity: theoretical algorithms broadly studied into machine learning field can be benefited by the amount of data collected by automatic speech recognition researchers, approaching machine learning to the reality; at the same time, automatic speech recognition area needs algorithms to solve complex problems, and they can be extracted from machine learning research area [Deng and Li, 2013].

In particular, language recognition, as a part of automatic speech signal processing field, can be viewed as a classification task, where the input data are segments of speech that want to be classified into different classes (languages). Thereby, machine learning algorithms can provide the best classifier to solve this task. Among the amount of options that can be chosen to solve the problem, deep neural networks provide, not only a classifier but a form to extract features from input data, in an automatic way, that represent them properly for classification purposes.

The two main topics of this Master Thesis, language recognition and deep neural networks, will be introduced in this chapter. After that, the motivation and goals of the Master Thesis will be presented and an outline of the dissertation will be included at the end of this section.

### 1.1. Language Recognition

Within voice signal processing field, a big amount of tasks may be included. Three of them can be highlighted due to their relation with this Master Thesis: 1) Speech Recognition, 2) Speaker Recognition and 3) Language Recognition. All of them have properties in common, such as, for example, a phase of feature extraction at the beginning of the system. However, their objectives are completely different. Then, let's introduce briefly each of them:

#### 1. Speech Recognition

This task deals with the problem of determining what the message in a segment of voice signal is, *what is being said*. To solve this task, it does not matter who the speaker is, so the speaker can be considered as a source of variability. As it will be shown, each mentioned

task has its own factors of variability to deal with, and which make the problem difficult to solve.

### 2. Speaker Recognition

When we talk about speaker recognition we mean the task of recognizing the speaker's identity, *who is speaking*. In this case, the content of the message itself can be considered a variability factor that the system should deal with (if we are referring to text-independent systems).

### 3. Language Recognition

It is known as language recognition the task of determining *which language is being used* in a segment of speech. As in the other tasks involved in speech processing, there are many variability factors that affect at the system performance as, for instance, the message itself and its speaker. This means that language identification should be a speaker and text independent task.

As it was mentioned before, all these tasks have properties in common, but this Master Thesis will be related to the problem of language recognition, so, this issue will be explained deeper.

First of all, language recognition can be referred to spoken or written resources. In our case, we will always refer to the spoken language case. As the goal of the task is to identify which language is being spoken in a segment of speech, the task can be described into two phases: create a model of the signals for each language and measure how similar an input signal is with the model.

The problem of language recognition (or spoken language recognition, SLR) can be broadly divided into different variants:

- Language verification: a decision about whether a target language is spoken in an input signal or not. The system outputs often a likelihood (plausibility or confidence measure) of the decision as well.
- Language identification in a closed-set: the output of the system for this modality is which language, among N given languages, is spoken in a given input signal.
- Language identification in an open-set: this variant of the task is similar like the close-set modality, but the system can output none of them as a result.

As far as the structure of a SLR system is concerned, it can be divided into different parts:

- Feature extraction: in this part, the system will extract some features from the input signal, ideally, those that give us relevant information for classification purposes. Thus, a new representation of the input signal is obtained.



**Figure 1.1:** General structure of a language recognition system.

- **Classification:** this phase is based on giving a score (confidence or distance measure) in order to accomplish the classification task.
- **Decision:** given the scores obtained in the previous part, the system makes a decision, depending on the task (verification or identification).

Finally, regarding the applications of language identification systems, we can mention audio indexing, information retrieval and call center monitoring. Furthermore, these systems can be used for filtering telephone calls and retaining only those in the language of interest, or for preprocessing the input speech signal in multilingual dialog systems [Ambikairajah *et al.*, 2011; Gonzalez-Dominguez *et al.*, 2010].

## 1.2. Deep Neural Networks

Deep Neural Networks (hereafter, DNNs) are a part of the machine learning field whose success is relatively new. Under this name, a range of structures is included and all of them have something in common: taking classical neural networks (the shallow ones, which are those whose structure is composed of just one hidden layer) as a starting point, hidden layers are added so that they allow dealing with complex problems within the machine learning field, in which traditional structures are limited.

Thereby, DNNs try to emulate the complex human learning system, extracting features at multiple levels of abstraction and learning complex functions directly from the input data, without depending completely on human-crafted features. The ability to automatically learn powerful features is becoming increasingly important as the amount of data and range of applications to machine learning methods carries on growing [Bengio, 2009].

However, successful experimental results using deep architectures with more than one or two hidden layers were not reported until 2006 [Bengio, 2009] (except for the case of convolutional networks, that will be described in section 2.2.5) due to limitations in training this kind of structures. Some of these limitations were not coming from a theoretical point of view, but they were due to limitations in hardware devices. Moreover, although algorithms to train them already existed, the random initialization of the parameters makes these architectures yield poor results [Bengio *et al.*, 2007; Erhan *et al.*, 2009].

All these limitations were solved with the evolution of hardware devices and with a learning algorithm that greedily trains one layer at a time [Hinton *et al.*, 2006], taking advantage of

unsupervised learning algorithms to initialize the parameters.

Nevertheless, the huge number of free parameters to train in this kind of neural networks is one of the main disadvantages that they have. Moreover, other of their drawbacks that it can be highlighted is the computational cost that their training algorithms present and the amount of data that should be used to train the whole architecture.

### 1.3. Motivation

The idea of combining DNNs and SLR areas comes from the good results obtained in the speech recognition field by using this kind of structures [Hinton *et al.*, 2012a; Jaitly *et al.*, 2012; Lee *et al.*, 2009b; Mohamed *et al.*]. Those results make us think that deep architectures have the capability to obtain a good speech signal modeling that can be useful for the language recognition task.

On the one hand, as it has been mentioned above, it has been numerous the researchers within machine learning field that have obtained better performance on systems using DNNs than other architectures. This has happened, not just in the speech processing area but in other fields such as, for instance, computer vision tasks [Poon and Domingos, 2011; Salakhutdinov and Hinton, 2009]. All these previous works give us an idea of how powerful DNNs can be as a machine learning tool.

On the other hand, the SLR problem shares most of its issues with other related research tasks such as speech and speaker recognition, so the solution to this task can be extrapolated to these fields. For instance, the inter-session variability problem, which is the set of differences between segments of speech that are not related to the language or the speaker, has consequences that difficult the main task of language or speaker recognition themselves.

Moreover, the study of both areas and the systems developed in this Master Thesis can be ported to other relevant research fields such as, for example, biometric recognition systems based on other traits (face, signature, fingerprints, iris).

### 1.4. Goals of the Master Thesis

The main goal of this Master Thesis is to apply learning algorithms based on deep neural networks to the problem of language recognition.

Thus, the Master Thesis can be divided into two parts:

- Theoretical framework: This part includes the study of different architectures of deep neural networks and their properties to know which of them are suitable for the problem to be solved. In addition to this, language recognition systems based on acoustic models have been studied in order to enlarge the knowledge of the problem itself and the methods that can help to solve it.

- Experimental framework: The aim of this part is the development of a SLR system based on DNNs. As a starting point, some experiments with handwritten digits have been replicated, and, after that, the system has been adapted to cope the SLR problem.

## 1.5. Outline of the Dissertation

The Dissertation is structured as follows:

- Chapter 1 introduces the issues of language recognition and deep neural networks and gives the motivation, objectives and outline of this Master Thesis.
- Chapter 2 summarizes the state of the art in language recognition and deep neural networks, the main issues discussed in this Dissertation.
- Chapter 3 presents applications of deep neural networks which encourage this work and describes the proposed method of language recognition by using deep neural networks.
- Chapter 4 describes the experiments performed during this work, detailing and analyzing the results achieved.
- Chapter 5 summarizes the main conclusions drawn from this work, outlining also future research lines.



## Chapter 2

# Related Works and State of the Art

THE TWO AREAS IN WHICH THIS WORK IS FOCUSED are language recognition and machine learning based on deep neural networks. Both of them have been widely studied by researchers during the last three decades.

This chapter provides a brief overview of both fields.

First, in section 2.1, some reasons that make language recognition an important field for researchers will be shown, as well as a summary of the two main approaches on which language recognition systems are based: spectral or acoustic approaches and phonotactic systems.

Second, section 2.2 will present how deep neural networks have been arisen as a new paradigm within the machine learning field lately, and will also summarize some concepts related to machine learning such as different types of learning, architectures and training algorithms.

### 2.1. Automatic Language Recognition Systems

Among the amount of topics where automatic language recognition systems are involved, this section will describe briefly the advantages of studying those systems, and will present some ideas on which the most common approaches of language recognition systems are based.

All the concepts that will be mentioned in this work are referred to systems that receive a speech signal as input, that is, they try to figure out which language is spoken in a concrete segment of speech. They are known as spoken language recognition (SLR) systems too, to be distinguished from those that consider text as their input.

#### 2.1.1. Why Language Recognition?

As it was mentioned before, SLR area belongs to the huge field of speech processing, and is closely related to other parts of it such as speaker and speech recognition.

One of the main problems that all these tasks share is the variability between different sessions. As far as the language recognition task is concerned, that is the problem that, although the spoken language in two utterances is the same, their content can be completely different,

and this can cause a bad performance of the system if it classifies the utterance into different languages, making a mistake in the final decision.

Apart from the variability problem, SLR shares the same kind of input data with other tasks involved in speech processing. This input data, that are segments of speech, are complex signals that usually need to be preprocessed and, thus, features which are relevant for the purpose of the final system are not easy to extract. The extraction of features from input data is other property that the language recognition task has in common with speech and speaker recognition problems.

However, the final aim of a SLR system is different from, for example, a speaker recognition one. The former tries to classify into known classes and, normally, the classes to train and test the system are the same. On the contrary, if we talk about a speaker recognition system, the speaker that the system wants to recognize can be different from those that have been used to train it, making the task even more challenging.

Moreover, the task of SLR can be viewed as a classification task, where those features that are relevant to discriminate into different classes should be extracted from the input data, sharing, indeed, properties with lots of problems within the machine learning field.

Other tasks related to the one that this work is focused on are those within image processing field such as handwritten digit recognition or object identification, in which the image should be modeled to extract the important information to achieve the objective of the system.

Finally, the large amount of applications that a SLR system can have in actual life, such as those mentioned in section 1.1, make it an interesting problem to start with.

### 2.1.2. Spectral Systems: Acoustic Approaches

Spectral systems, also known as acoustic or lower level systems, are one of the approaches that it can be used to face the problem of SLR.

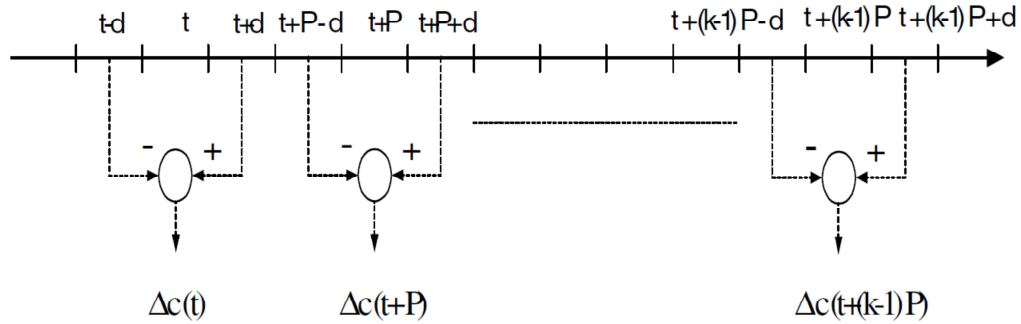
These systems are based on spectral features extracted, typically, from short segments of speech (normally, around 20 or 30 milliseconds long).

The most common acoustic features used to represent the speech signal are the Mel Frequency Cepstral Coefficients (MFCCs), but, to make the representation more compact, the first and, depending on the application, the second derivatives are appended to the static information provided for the MFCCs. Sometimes, prosodic features such as duration, pitch or stress are used to complement the representation of the signal.

However, in SLR in particular, the Shifted Delta Cepstrum (SDC) representation is commonly used since it was proposed in Torres-carrasquillo *et al.* [2002]. The SDC representation consists of a sequence of delta features computed at different time instants and is defined by the following 4 parameters (N-d-P-k):

- N corresponds to the number of cepstral coefficients (static features) computed at each frame.





**Figure 2.1:** Computation of the SDC feature vector at frame  $t$  for parameters  $N$ - $d$ - $P$ - $k$  [extracted from Torres-carrasquillo et al. [2002]].

- $d$  represents the time advance and delay for the delta computation, that is, the length of the window used to estimate the delta.
- $P$  is the time shift between consecutive deltas.
- $k$  is the number of blocks whose delta coefficients are concatenated to form the final feature vector.

This idea is illustrated in Figure 2.1.

Consequently, the length of the final feature vector would be  $kN + N$ , since the SDC representation is often concatenated with the static features. For instance, if the chosen configuration were 7-1-3-7, the final vector would be composed of 56 components, which is the case of some experiments shown in chapter 4.

Regarding the classifiers themselves, many different approaches have been used, since the simplest ones based on Gaussian Mixture Models (GMMs) to those that use Support Vector Machines (SVMs), Factor Analysis (FA) or Total Variability (TV) [Ambikairajah *et al.*, 2011].

### 2.1.3. High Level Systems: Phonotactic Approaches

Other successful group of approaches in the SLR field is the set of systems that consists of those based on high level information that can be extracted from the speech signal.

These high level systems are also known as phonotactic systems due to this kind of approaches takes into account the restrictions on the possible combinations between phonemes in a given language. Thus, an important part of this kind of systems is the phonetic recognizer that processes the speech signal and converts it into a sequence of tokens that identifies the phonemes. It can be used just a single phonetic recognizer or several ones in different languages (this is the

case of the parallel phonetic recognizer) to improve the performance of the system [Zissman and Singer, 1994].

Then, phonotactic systems take advantage of information about the phonetic distribution, the morphological rules that are defined in a language and other information related to the language itself.

Instead of considering the whole words, classical approaches use phonemes, which is easier for the system. Thereby, these systems utilize statistical language modeling techniques to model the frequencies of appearance of phones in a particular language. They also take into account the phone sequences called n-grams for a concrete language, that given the grammar of that particular language, a probability of the following phoneme given the n previous ones can be defined as follows:

$$P(s_n | s_{n-1}, s_{n-2}, \dots, s_1) = \prod_{i=1}^n P(s_i | s_{i-1}, s_{i-2}, \dots, s_1)$$

Due to the large amount of combinations that can result from phonemes that make up a particular language, this kind of models are usually reduce by taking into account just one, two or three previous phonemes:

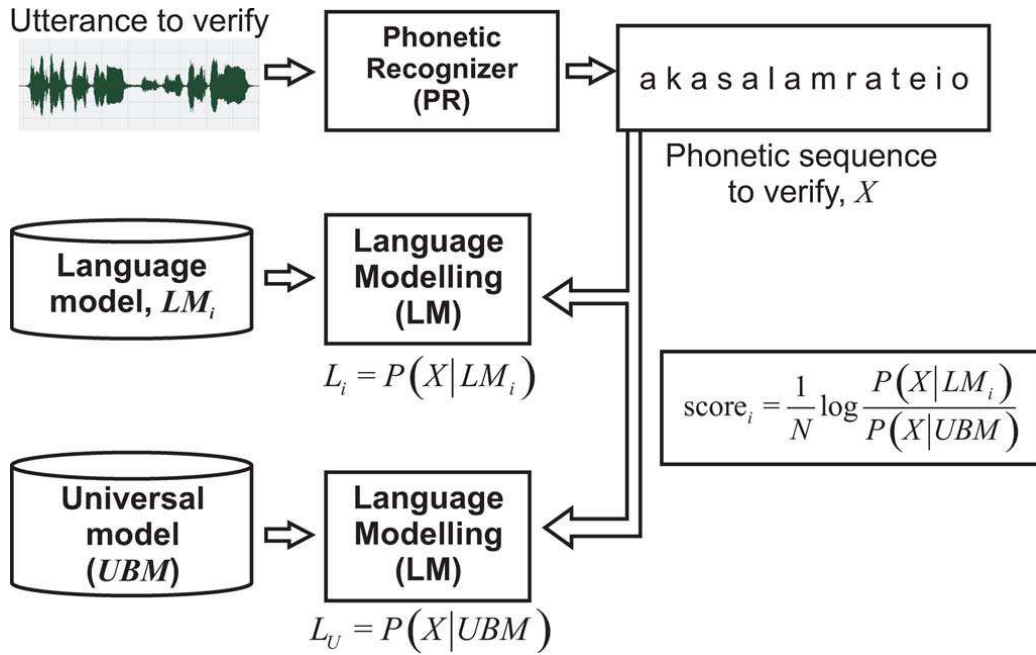
- Trigrams:  $P(s_n | s_{n-1}, s_{n-2}, \dots, s_1) = P(s_n | s_{n-1}, s_{n-2})$
- Bigrams:  $P(s_n | s_{n-1}, s_{n-2}, \dots, s_1) = P(s_n | s_{n-1})$
- Unigrams:  $P(s_n | s_{n-1}, s_{n-2}, \dots, s_1) = P(s_n)$

This combination of a phonetic recognizer and a language model is the underlying idea in the case of the phone recognition language modeling (PRLM) approach, widely use within the field of language recognition. Figure 2.2 illustrates its structure for a verification task, which is given an utterance, the system outputs a high score whether the utterance belongs to the target language or low score otherwise.

Hence, the stages that a PRLM system has to carry out are the following ones:

- Training a phonetic recognizer for some language.
- Training a Universal Background Model (UBM) for all the languages.
- Training a model for each language, often by adapting the UBM.
- Recognizing the language by comparing the utterance with the UBM and the language models. This phase outputs a score that it should be used to make the final decision in the case of a verification task.

As it was mentioned before, more than one phonetic recognizer can be used, fusing the results of each individual phonetic recognizer. This results in a new system known as Parallel PRLM (PPRLM), that achieves better performance than a PRLM system but by increasing the



**Figure 2.2:** Verification task in a PRLM system [extracted from Gonzalez-Dominguez et al. [2010]].

complexity of the system and, hence, the computational cost. These PPRLM schemes are still a key in the state-of-the-art SLR systems.

Other alternatives based on the same information than PRLM and PPRLM systems are the Phone Support Vector Machines (Phone-SVMs), which use SVMs for classifying the whole n-gram probability matrices, instead of using them in a likelihood ratio framework [Campbell et al., 2004].

Finally, a complete SLR system is usually a combination of different subsystems, so a fusion module is frequently required. This part of the system is commonly known as the back-end module, and it can be considered as a *pre-calibration* stage [Gonzalez-Dominguez et al., 2010]. However, although these back-end strategies are really important in all the systems, they will not be explained in this work due to it has not been used in the development part.

## 2.2. Deep Neural Networks

This section is intended to provide an overview about how deep architectures have emerged as a new paradigm in recent years and why they have become a breakthrough within the machine learning field, transferring, at the same time, their influence to other interesting research fields as it can be the speech processing area.

Furthermore, some basic concepts about learning algorithms will be summarized, such as the differences between what is called supervised and unsupervised learning algorithms, things that make shallow and deep architectures different, and, finally, some ideas about how to train

deep architectures unlike the shallow ones.

### 2.2.1. A New Machine Learning Paradigm

Broadly speaking, the main aim of artificial intelligence could be stated as “make computers be able to model our world” [Bengio, 2009]. That implies to process a large quantity of information to answer questions and generalize to new contexts. Here, it is where machine learning algorithms play an important role. During lots of years, much progress has been made in this field, but the challenge of the artificial intelligence field still remains: computers can not understand well enough real scenes or describe them in natural language [Bengio, 2009].

One of the reasons which make human brain and learning algorithms performance so different is the way both extract useful information from the data. It is believed that human brain acts as a feature extractor that, gradually, gets information from the data at different levels of abstraction [Serre *et al.*, 2007]. In other words, it tries to decompose the problem to solve into some sub-problems with less complexity, obtaining, at the same time, different levels of representation.

This behavior is what machine learning structures and algorithms have tried to imitate. Firstly, the system captures low-level features that are invariant to small variations (for example, geometric variations if we talk about a computer vision task), transforms them to increase their invariance, and, finally, extracts useful information, which means frequent patterns that could be generalized to other data.

All this process of extracting useful information from raw input data makes necessary to have a structure with the ability of transforming the input in a non-linear way. In these terms, learning algorithms should be able to apply mathematical transformations or functions that are highly non-linear and varying.

Most of structures that have been used during years within the machine learning field have not enough capability to apply those complex functions that are necessary to solve certain problems. Classic architectures have usually one hidden layer that implies just one non-linear transformation of the data. The idea of deep architectures comes from this point: if more transformations are required, more hidden layers will be stacked. But this has many consequences for learning algorithms, such as the more complex the function to model is, the more local minimum can be found.

Thus, training algorithms that had been used for many years, did not work with these new architectures, known as “deep schemes” due to their higher number of layers respect to the previous ones. Backpropagation and gradient descent algorithms did not yield good results in experiments that used these new architectures, since weights and other parameters converged into really small values, close to zero.

There was an exception: Convolutional Neural Networks. This deep architecture yielded the first successful results when Yann Lecun used his structure known as “LeNet” for classification tasks [LeCun *et al.*, 2001]. In this architecture, the amount of free parameters that have to be learnt is reduced thanks to many of those parameters are shared between different parts of the

network. However, these networks work in a supervised way, so they need a huge amount of labeled data to train, which can be considered a drawback for some problems where it is not easy to get these labels.

From the point of view of unsupervised learning, it was not until 2006, when successful results were achieved by Hinton et al. at University of Toronto with which is called Deep Belief Networks (DBNs). This kind of networks was trained one layer at a time, taking advantage of unsupervised learning algorithms, which are the Restricted Boltzmann Machines. These networks exploit the advantages of unsupervised learning, which make them suitable for problems where lots of unlabeled data are available, such as audio tasks. Consequently, successful improvements have been achieved in the speech recognition field, as it was reported in Hinton *et al.* [2012a] and Mohamed *et al.*, field closely related to this Master Thesis.

Other consequence of augmenting the complexity of the network is the increase of the number of free parameters that has to be estimated or learnt. This means that computers need more capacity of computation, and more data to store. Thus, the limitations of hardware resources have proved that, although theoretical algorithms existed and were correct before 2006, they could not have been exploded. Fortunately, those limitations have been got over nowadays, although the high computational cost that DNNs present is still an unsolved problem.

There are more open issues concerning DNNs, as, for instance, how much knowledge of a certain problem the network should have or how to choose the parameters or structure of the network that suit a concrete problem. As an example, theoretical results have shown that there is no universally right depth for the network so that selecting one depth or other one is a problem dependent issue [Bengio, 2009].

However, although these open issues can be considered as big disadvantages of DNNs, they can be seen as an emerging research field with lots of topics to study as well.

To conclude, all the advantages that have been presented, the huge amount of applications where deep neural networks have achieved successful results, among speech and speaker recognition tasks are included, and the unsolved issues that these schemes have, give us reasons to consider deep architectures as an interesting new machine learning paradigm to explode.

### 2.2.2. Supervised and Unsupervised Learning

A learning algorithm can be defined as a form to calculate a prediction (output) from input data. This prediction is, indeed, a function that matches inputs with outputs.

The concept of learning itself is related to generalization. In a classification task, this can be defined as the ability to categorize correctly new examples that differ from those used for training. Thereby, datasets used in machine learning tasks should be split into, at least, two parts:

- Training data: This set covers the samples used to estimate the parameters of the objective function, with the objective of minimizing the error, for instance, the misclassified examples in a classification task.

- Test data: It is composed of examples that let prove the system. These examples are not included in the training dataset and, thus, they do not interfere in the parameters selection.

Regarding learning algorithms, they can be divided into two main groups: 1) supervised or 2) unsupervised algorithms, depending on the information that the system uses for training (labeled or unlabeled data). There are also semi-supervised algorithms and what is known as reinforcement learning, but we will focus on the first two cases.

### 1. Supervised learning

If we talk about a classification task, which is a machine learning problem where the output of the system should be the class or category of a certain input, supervised learning includes algorithms that use the information of the class to train. This modality requires which is called labeled data. In this manner, the cost function to minimize can be an error function that measures the difference between the predicted class and the actual class of a certain input.

Among this kind of algorithms, it can be highlighted Logistic Regression, Multilayer Perceptron and Deep Convolutional Network, due to their relation with this work.

### 2. Unsupervised learning

This set of algorithms is also known as algorithms based on clustering. The most known example is the K-means algorithm that tries to create groups among the data based on some measure of similarity between them.

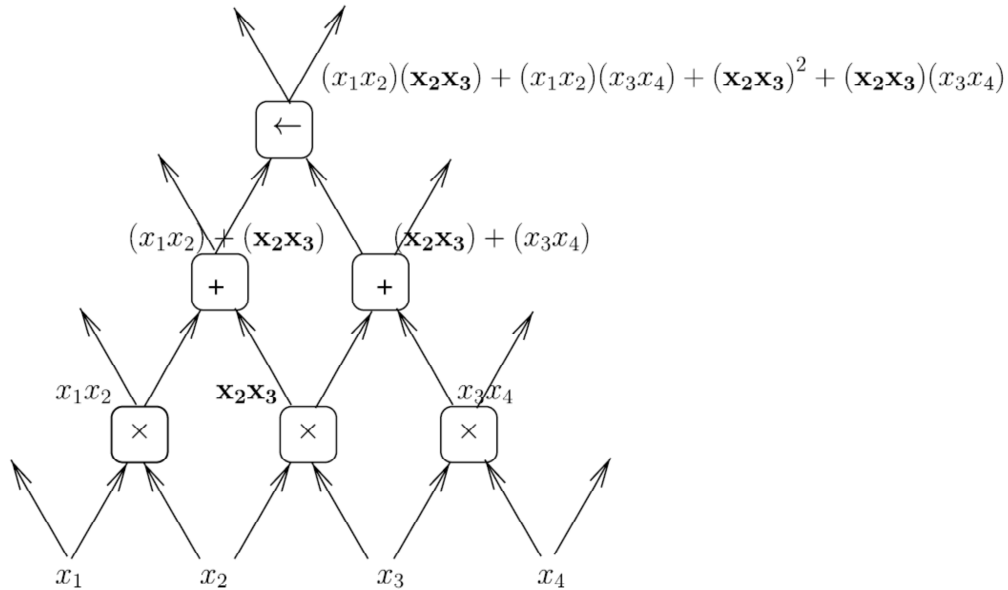
These algorithms can be understood as a way of modeling the distribution of the input data too. From this point of view, unlike supervised learning methods, they cannot measure an error between the predicted class and the actual class since they do not have the information about the actual class. Instead, they can try to minimize a “*reconstruction error*”, and this may be the cost function that the algorithm will consider to train.

Within this set, Auto-Encoders and Restricted Boltzmann Machines (RBMs) are included.

Apart from these two groups, due to its relation with this work, we can mention a semi-supervised learning algorithm: Deep Belief Networks [Hinton *et al.*, 2006]. As it will be shown in subsection 2.2.4, this architecture uses RBMs as building blocks, so they are trained in an unsupervised way. But its training has also a “fine-tuning” phase, that constitutes a supervised learning algorithm.

## 2.2.3. From Shallow to Deep Architectures

As it was commented before, from a theoretical point of view, shallow architectures present limitations that deep schemes can potentially solve.



**Figure 2.3:** Expression computed by a deep architecture with sums and products [extracted from Bengio [2009]].

The main motivation to explore deep architectures lies on the idea that some functions cannot be efficiently represented, which means with a reasonable number of parameters to tune by learning, by architectures that are too shallow. When this happens, it is said that a function is *compact* [Bengio, 2009]. Thus, a function that can be represented in a compact form by certain architecture may carry an exponential increase in the number of parameters to tune if other architecture with one layer less is used instead.

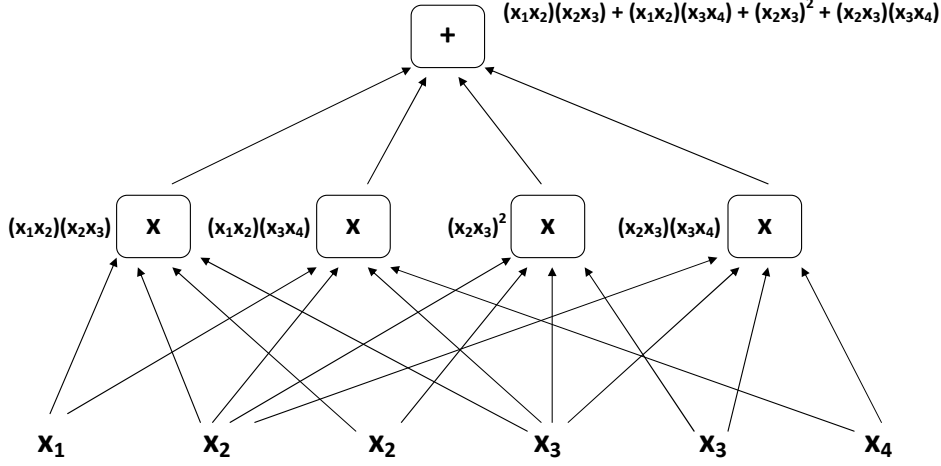
All this implies that, if the amount of data available to train is not enough with respect the number of parameters to be tuned, the representation that will be achieved by a shallow network can be not as good as the representation that a deep architecture could accomplish.

Figure 2.3 extracted from Bengio [2009] shows an example of an expression where the term  $x_2x_3$  occurs more than once, and how a deep architecture can avoid repeating that computation many times.

With the structure shown in figure 2.3, the number of parameters or connections to tune by learning would be 12, while with the shallow structure represented by figure 2.4 the amount of parameters increases up to 20.

Although all the advantages that deep architectures seem to present versus the shallow structures, the big amount of variations within input data from a certain problem, for example data from a vision or audio context, is still a problem for machine learning algorithms.

Also, it should be taken into account that not every problem in the world is so difficult that a deep architecture is required to solve it, so that the simplest structure that fits the problem should be used in those cases.



**Figure 2.4:** Same expression from figure 2.3 computed by a shallow architecture with sums and products.

Regarding the issue of how to build a deep architecture, some ideas about it will be presented below, according to the software and architectures [LISA] used in the experimental part of this work.

In that sense, shallow structures can be seen as building blocks for deep ones. For example, if a classification task based on supervised learning is being implemented, the structure shown in figure 2.5 could be used. That architecture is composed of the next elements:

- An input layer, which just represents the input used to feed the network.
- Hidden layers, that apply non-linear transformations to the input data, such as, for example, a *sigmoid* or *tanh* functions defined as follows:

$$\text{sigmoid}(a) = \frac{1}{1 + e^{-a}}$$

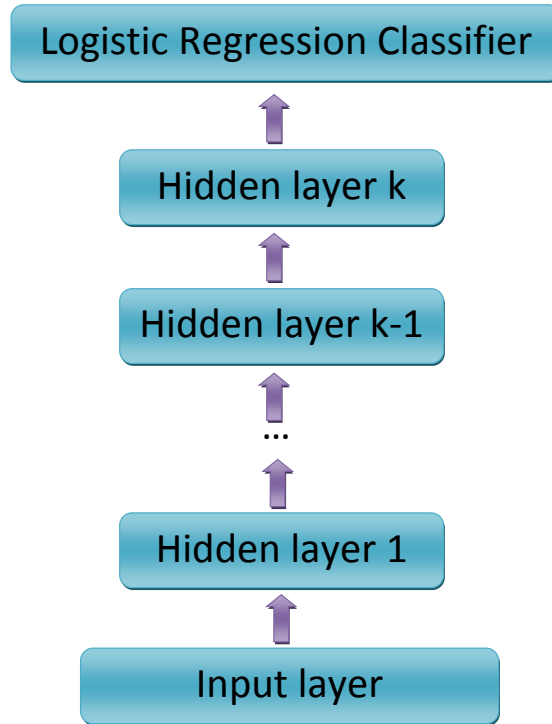
$$\text{tanh}(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}}$$

It can be added as much hidden layers as it was necessary for the problem to solve, building a structure where the input of one hidden layer is the output of the previous one.

- A logistic regression classifier, which is the output layer and performs the classification maximizing a cost function, as, for example the one defined as follows:

$$P(Y = i|x, W, b) = \text{softmax}_i(Wx + b) = \frac{e^{W_i x + b_i}}{\sum_j e^{W_j x + b_j}}$$





**Figure 2.5:** Example of deep architecture for classification.

Where the selected class for a certain input would be:

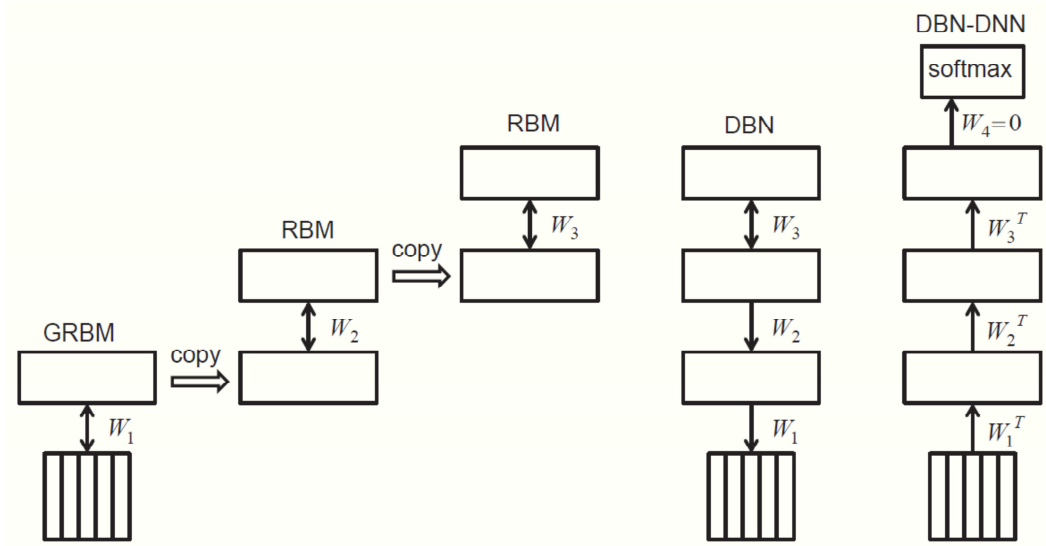
$$y = \operatorname{argmax}_i P(Y = i | x, W, b)$$

The parameters to be tuned, in that case, would be the *weight* matrix  $W$ , and the *bias* vector  $b$ , that can be learnt by using, for instance, a gradient descent algorithm [Bishop, 2007].

Following the same idea, but using a semi-supervised learning algorithm, Deep Belief Network (DBN) constitutes an architecture where Restricted Boltzmann Machines (RBMs) are used as building blocks for pre-training [Hinton *et al.*, 2006]. Finally, a logistic regression classifier can be added as output layer to perform the same classification task as the one mentioned before. The algorithm widely used to train this DBN architectures will be summarized in section 2.2.4.

#### 2.2.4. Training Algorithms for Deep Architectures

According to some experimental results [Bengio *et al.*, 2007; Erhan *et al.*, 2009], training deep architectures in the same way as shallow structures are trained does not work as it is expected: high training errors and poor generalization are usually obtained when a random initialization of the parameters is used.



**Figure 2.6:** Training algorithm for a deep belief network [extracted from Hinton et al. [2012a]].

Except for the case of convolutional deep neural networks (see section 2.2.5), training algorithms for shallow architectures had to be modified to be used with deep architectures.

The solution discovered to solve the problem of poor generalization obtained by deep architectures trained as shallow architectures was the use of unsupervised learning for pre-training [Hinton *et al.*, 2006] since, before this, the parameters were initialized randomly. With that phase of pre-training, the structure of the input data is discovered in an unsupervised way, favoring generalization. After that phase, a fine-tuning process is applied to adjust the parameters for the final purpose of the system. In this sense, pre-training can be viewed as a regularization source.

It was also shown that pre-training helps to get better optimization and avoids the algorithm to get stuck in local minima that are far from the global minima.

An example of this algorithm composed of two stages, pre-training and fine-tuning, is that used to train DBNs. In this case, each RBM can be trained with the Contrastive Divergence algorithm in the pre-training phase. Then, the stack of RBMs can be converted to a single generative model, by replacing the undirected connections by top-down, directed connections. Finally, the resultant DBN can be discriminatively trained to the final objective (classification, for example). This process is illustrated by figure 2.6.

### 2.2.5. Case of Success: Deep Convolutional Networks

As it was mentioned before, DNNs did not accomplish good performance due to the difficult that was found before the unsupervised *pre-training* stage. However, convolutional deep neural

networks (convolutional DNNs) [LeCun *et al.*, 2001] are an exception.

These networks are models based on the structure of the visual system and are composed of two kinds of layers: convolutional layers and subsampling layers. The first ones act as a feature extractor where each unit is connected to a local subset of units in the layer below. Some units that are related because of their location share their parameters, allowing the network to extract the same features from different locations in the input, and reducing the amount of parameters to tune. Subsampling layers reduce the size of the representations obtained by convolutional layers by applying a subsampling operation, and making the network, in some way, invariant to small translations and rotations [Bengio, 2009].

Moreover, convolutional nets can be trained as a classic *feedforward* network, by using, for instance, supervised learning based on gradient descent algorithms [LeCun *et al.*, 2001]. Even with random initialization of the parameters (*weights*), convolutional DNNs perform well, not just in tasks such as digits recognition on MNIST [LeCun *et al.*, 2001], but also in object classification tasks on Caltech-101 [Ranzato *et al.*, 2007].

According to Bengio [2009], one hypothesis about why this kind of networks works well even when they are trained with gradient-based algorithms is the small number of inputs per neuron that makes the gradient not so much diffusing to be useless. In this article, it is mentioned too that the hierarchical local connectivity structure of convolutional networks makes the set of parameters be in a favorable region where gradient-based optimization works.

Thereby, all the examples where convolutional DNNs achieve good results make them be among the best pattern recognition systems [Bengio, 2009], and this is why the experimental part of this Dissertation is performed using this type of deep architectures.



## Chapter 3

# Deep Neural Networks Applied to Language Recognition

DEEP NEURAL NETWORKS, as a machine learning tool, can be applied in many research fields, in which their influence is being increased currently.

First of all, it is important to highlight the results obtaining by these deep architectures in two classification tasks within the computer vision field: 1) handwritten digit recognition and 2) object classification.

### 1. Handwritten Digit Recognition

It is a very standard task to test machine learning algorithms, frequently using the MNIST database [Lecun and Cortes].




































Although results obtained with classical machine learning structures are pretty good, the test error rate can be reduced by using deep neural networks [Ciresan *et al.*, 2010; LeCun *et al.*, 2001].

Figure 3.1 shows the 35 misclassified examples out of 10.000 test cases obtained with one of the deep architectures explained in Ciresan *et al.* [2010]. In the figure, it is also shown the two most likely predictions for each example (bottom) and the correct label according to MNIST (top). It should be mentioned that the correct answer is the second guess of the network for 30 out of the 35 misclassified digits.

### 2. Object Classification

Other application where DNNs have shown their powerful properties is the object classification task on challenging datasets such as ImageNet database [Deng *et al.*, 2009].

Figure 3.2 shows some examples of predictions obtained with Convolutional Neural Networks as object classifiers by Krizhevsky *et al.* [2012].

 1 2 1 7	 1 1 7 1	 9 8 9 8	 9 9 5 9	 9 9 7 9	 5 5 3 5	 8 8 2 3
 4 9 4 9	 5 5 3 5	 9 4 9 7	 4 9 4 9	 4 4 9 4	 0 2 0 2	 5 5 3 5
 6 6 1 6	 4 4 9 4	 0 0 6 0	 6 6 0 6	 6 6 8 6	 1 1 7 9	 1 1 7 1
 9 9 4 9	 0 0 5 0	 5 5 3 5	 8 8 9 8	 9 9 7 9	 7 7 1 7	 1 1 6 1
 2 7 2 7	 8 8 5 8	 2 2 7 8	 6 6 1 6	 6 5 6 5	 4 4 9 4	 0 0 6 0

**Figure 3.1:** The 35 misclassified digits [extracted from Ciresan et al. [2010]].

These cases-of-success give us an idea about how powerful DNNs can be as a machine learning tool. But, what mainly motivate this Master Thesis are the good results that have been achieved with the application of DNNs to speech processing tasks.

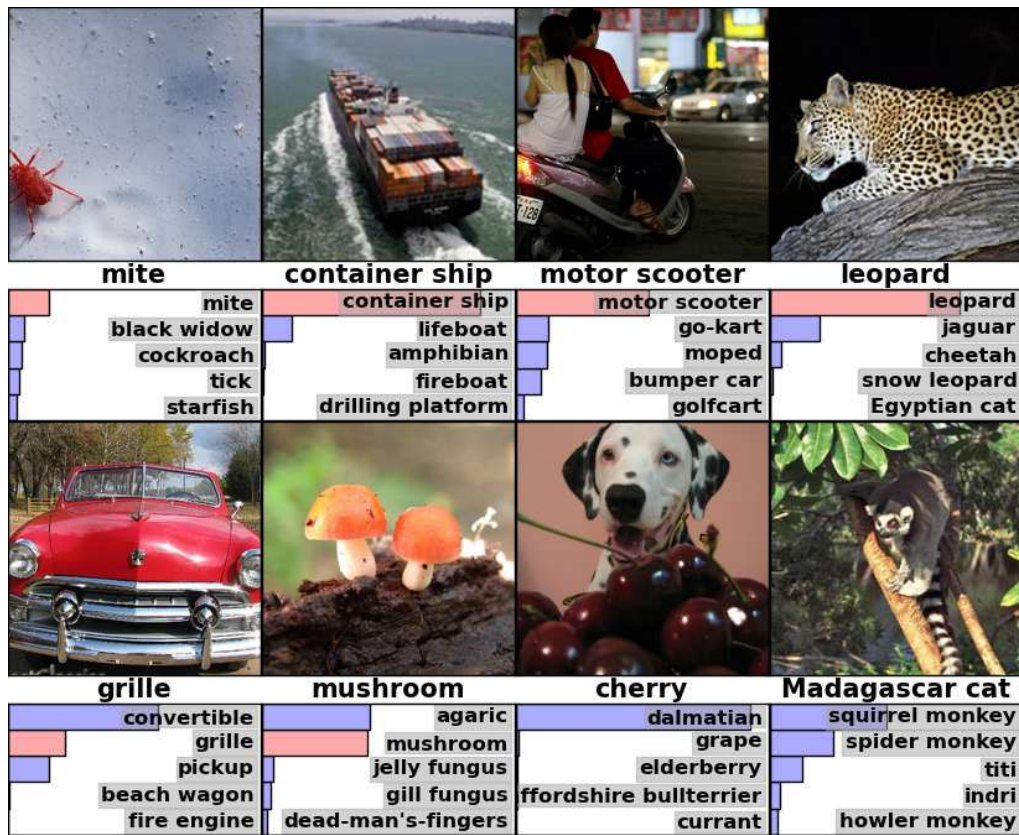
The reminder of this chapter is organized as follows. Some results obtained using Deep Belief Networks (DBNs) in the speech recognition field will be exposed. Then, some ideas about how DNNs could be applied to the SLR problem will be proposed and architectures that have been used in the experimental part of this works will be presented, as well.

### 3.1. DNNs Applied to Speech Recognition

As it was mentioned before, the motivation of this work comes from promising results obtained with the application of different types of DNNs to the speech processing field [Deng and Li, 2013; Hinton *et al.*, 2012a; Lee *et al.*, 2009b; Mohamed *et al.*].

In particular, Hinton *et al.* [2012a] provides an overview about how speech recognition tasks can be improved by using DNNs for acoustic modeling, according to results obtained recently by four relevant researcher groups (University of Toronto, Microsoft Research, Google and IBM Research).

Traditionally, Gaussian Mixture Models (GMMs) and Hidden Markov Models (HMMs) have been used as the main framework for speech recognition tasks. Within that context, the properties of GMMs make them suitable for modeling the probability distributions over vectors of input features that are associated with each state of an HMM [Hinton *et al.*, 2012a]. GMMs are



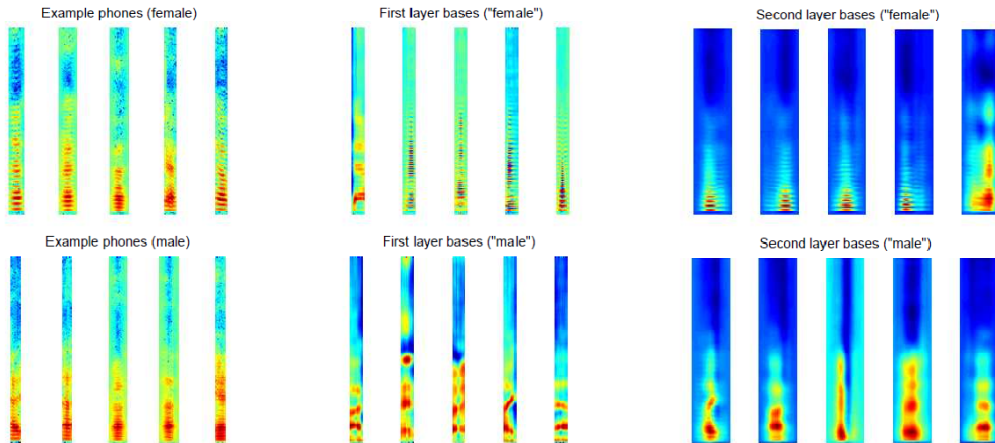
**Figure 3.2:** Examples of object classification predictions [extracted from Krizhevsky et al. [2012]].

able to model complex probability distributions if they have enough components, and can be easily fit to data with the Expectation-Maximization (EM) algorithm [Moon, 1996].

However, the GMM-HMM framework has some important limitations. For instance, GMMs require assumptions about the data distribution to estimate the posterior probabilities of HMM states. This problem does not occur by using DNNs instead of GMMs [Mohamed *et al.*]. But the main limitation that GMMs present is that “they are statistically inefficient for modeling data that lie on or near a nonlinear manifold in the data space” [Hinton *et al.*, 2012a].

Thereby, according to Hinton *et al.* [2012a], it is believed that there are models that exploit better the underlying information of the speech signal, by considering larger windows of frames, than GMMs.

DNNs have that capacity to learn complex models. Deep architectures used by the groups mentioned before take advantage of a generative pre-training stage, that tries to capture the data distribution, and, after that, the model is trained discriminately in what is called a “fine-tuning” stage.



**Figure 3.3:** Spectrograms of the same phoneme for female and male speakers and their representations obtained by a convolutional DBN [extracted from Lee et al. [2009b]].

Results reported in [Hinton *et al.*, 2012a] show that the DNN-HMM framework outperforms the traditional GMM-HMM benchmark in different speech recognition tasks (Phonetic classification on TIMIT, Switchboard, English Broadcast News, Bing Voice Search, Google Voice Input and Youtube), decrementing error rates in any case. It should be taken into account that, when enough data and components are available for a GMM-HMM based system, both DNN-HMM and GMM-HMM achieve similar results.

It is important to highlight that the computational cost of training algorithms to DNNs is still an open issue, since parallelizing their algorithms is not as easy as the EM algorithm [Hinton *et al.*, 2012a].

Other way of using DNNs in speech processing tasks can be as feature extractors: hidden layers of DNNs give different representations of the input signal that can be used as input features for classic GMM-HMM systems [Hinton *et al.*, 2012a].

In Lee *et al.* [2009b], convolutional deep belief networks (hereafter, convolutional DBNs) [Lee *et al.*, 2009a] are used to obtain a new representation of the input data, by training the structure in an unsupervised way. This also allows taking advantage of a big amount of unlabeled data available, while labeled data are not easy to find for some speech processing tasks.

In that work, MFCCs and the representation obtained in the first two hidden layers of a convolutional DBN are used as input for different systems and tasks, achieving better results than those obtained with just MFCCs as input. Figure 3.3 shows examples of spectrograms of the same phoneme for female and male speakers, and their correspondent representations obtained in the first and second hidden layers of a convolutional DBN. That can give us an idea of how the discriminative power of the input representation seems to be increased as we pass through the layers of the network.



All these ideas can be used in order to improve other speech processing tasks, such it can be the SLR problem on which this work is focused.

### 3.2. Proposed Method: DNNs Applied to Language Recognition

One of the ideas extracted from previous works (cited in section 3.1) that could be applied to the SLR field is the used of DNNs to extract features from the raw input data. In these terms, a new representation of the input signal could be obtained, and used in a classic system such as a Support Vector Machine (SVM). Among other things, this application would let us take advantage of an unsupervised learning stage and use lots of unlabeled data available in the world; indeed, labeled data would be needed just in the classic system to carry out the classification task.

However, this Dissertation proposes the use of DNNs as a complete system: feature extraction and classification is done by the DNN.

Concretely, convolutional deep neural networks have been the selected architecture to implement the experimental part of this work. This decision was based on the good results obtained with this kind of networks in different tasks [LeCun *et al.*, 2001; Lee *et al.*, 2009a,b]. Moreover, convolutional DNNs, due to their properties about sharing weights and pooling, usually have less free parameters to tune. This allows a better generalization even when the amount of available data to train is not as big as that which would be required to train other types of deep architectures.

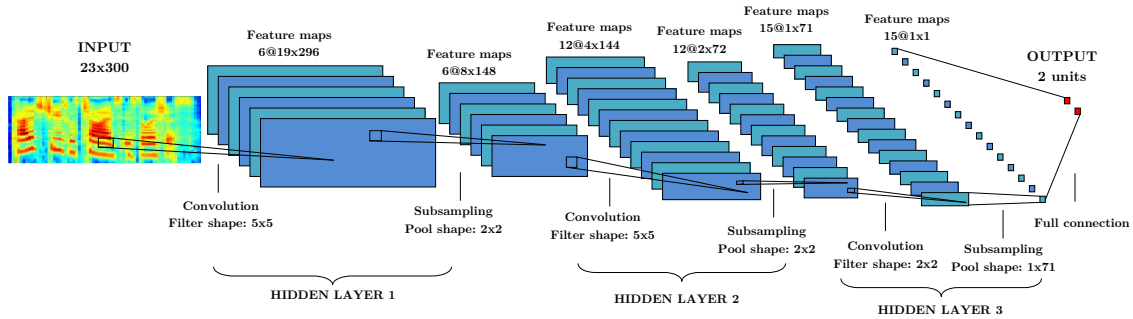
Therefore, taking the application of convolutional DNNs to the handwritten digits classification task as starting point, the idea was to adapt the structure to fit the problem of SLR.

Then, the process of that adaptation consisted of different stages, which are summarized below.

- Input

Firstly, it should be chosen what kind of input will be used to feed the network. In the handwritten digits recognition case, the input used is usually the gray-scale image that represents a certain digit [LeCun *et al.*, 2001]. Due to the properties of convolutional DNNs that make the relative position between pixels be important, the image is, indeed, a 2-dimensional representation of the signal. Likewise, the speech signal associated with a certain language is then represented by its spectrogram, i.e. a time-frequency representation.

Two kinds of representation have been used in the experiments carried out in this work: SDCs and Mel-scale filter-bank output. However, Mel-scale filter-bank outputs are used in most of the experiments. This is due to SDCs throw away some information in the sound wave that it is supposed to be irrelevant for discrimination [Mohamed *et al.*], but that process does not seem to be required when a DNN is being used.



**Figure 3.4:** Example of a convolutional DNN structure used in the experimental part.

As far as the time domain axis is concerned, segments of three seconds of speech are mostly used in the experiments, although there are some examples where the model used considers half a second of speech.

#### ■ Configuration of the network

One important issue of the use of DNNs is to choose the parameters to configure the structure of the network.

First of all, the number of layers selected for the experimental part of this work is five (or six):

- An input layer that just reshapes the input data into the required shape.
- Three (or four) convolutional layers (including the convolution and the max-pooling parts).
- An output layer that computes the score for each class. The number of output units will be given by the number of languages involved in each experiment.

Figure 3.4 shows an example of one of the structures used in the experimental part of this Dissertation.

Regarding the convolutional layers, some parameters (number of filters for each layer, filters shape and max-pooling shape) have to be set to different values, originating the set of experiments shown in Chapter 4.

#### ■ Labels

Similarly with what is done in the case of digits recognition, the labels used to identify a determined language are just an integer number (from 0 to the number of languages involved in the experiment).

- Training algorithm

The algorithm that has been selected to train the network in all the experiments is the stochastic gradient descent algorithm, based on minibatches. This algorithm uses a certain number of examples to estimate the gradient [LISA]. The minibatch size is set to 500 examples in all the experiments carried out in this work.

A technique of “early stopping” is used to combat overfitting: the performance of the model is evaluated on a validation set and, when the improvement is not relevant enough (depending on some configuration parameters), the algorithm stops.

The cost function that the algorithm tries to optimize (minimize in this case) is the negative log-likelihood (NLL), defined as follows:

$$NLL(\theta, D) = - \sum_{i=0}^{|D|} \log P(Y = y^{(i)} | x^{(i)}, \theta)$$

where  $D$  is the dataset,  $\theta$  represents the parameters of the model ( $\theta = W, b$ , weights and bias respectively),  $x^{(i)}$  is an example,  $y^{(i)}$  is the label corresponding to example  $x^{(i)}$ , and  $P$  is defined as the output of a *softmax* function as it was defined in section 2.2.3.

This is the same algorithm that is used in LISA by LISA lab (University of Montreal) for the case of handwritten digits classification.

- Output and Evaluation

The output of the last layer is considered a score, which is analyzed after. This allows extracting confusion matrices and error rates such as the Zero-One Loss (ZOL) or the Equal Error Rate (EER).



## Chapter 4

# Experiments and Results

THIS CHAPTER PRESENTS THE EXPERIMENTAL RESULTS obtained in this Dissertation.

The rest of this chapter is organized as follows. A reference system is described and it will be taken as a *baseline* to compare the results obtained by the proposed method. Then, the database and experimental framework involved in the experiments performed will be described. Finally, the results achieved will be detailed and analyzed.

### 4.1. Reference System

In order to have a baseline to compare with, one of the systems that ATVS - Biometric Recognition Group submitted to the NIST LRE 2009 evaluation has been tested for the same tasks performed with the proposed method of this work.

Concretely, the system that has been taken as reference system is a spectral system that consists of a GMM system with linear scoring and session variability compensation applied in the statistic domain, system named as *ATVS3* or Factor Analysis GMM Linear Scoring (FA-GMM-LS) in Gonzalez-Dominguez *et al.* [2010]. This system has been selected as reference system for this work due to it is among the state-of-the-art acoustic approaches in the SLR field.

The configuration of the system is as follows. The speech signal is represented by a parameterization consisting of seven MFCCs with CMN-Rasta-Warping concatenated to 7-1-3-7 SDC-MFCCs. Two Universal Background Models (UBMs) with 1024 Gaussian components were trained. One of them ( $UBM_{CTS}$ ) was trained with Conversational Telephone Speech (hereafter, CTS), data extracted from CallFriend, LRE05 and LRE07. The other one ( $UBM_{VOA}$ ) was trained with data from VOA (Voice of America radio broadcasts through Internet), provided by NIST. Regarding the amount of data used for training the UBMs, a total of 38.5 hours of speech were used in the case of  $UBM_{CTS}$  (about 2.75 hours per 14 languages). The  $UBM_{VOA}$  was trained with 31.2 hours in total, with 1.42 hours per 22 available languages.

Thereby, two different systems were developed, one for each UBM. Two session variability subspaces matrices were obtained ( $U_{CTS}$  and  $U_{VOA}$ ). The subspaces were initialized with PCA (Principal Component Analysis) based on Kenny *et al.* [2005]; Vogt and Sridharan [2008], taking

Prior model	Databases	# Languages	# Hours/language	Total
$UBM_{CTS}$	<i>CallFriend, LRE05, TrainLRE07</i>	14	2.75	38.5
$U_{CTS}$	<i>CallFriend, LRE05, TrainLRE07</i>	14	25	350
$UBM_{VOA}$	<i>VOA</i>	22	1.42	31.2
$U_{VOA}$	<i>VOA</i>	22	25	550

**Table 4.1:** Data used to train the elements of the reference system [extracted from Gonzalez-Dominguez et al. [2010]].

Amharic	Bosnian	Cantonese	Creole	Croatian	Dari
English (American)	English (Indian)	Farsi	French	Georgian	Hausa
Hindi	Korean	Mandarin	Pashto	Portuguese	Russian
Spanish	Turkish	Ukrainian	Urdu	Vietnamese	

**Table 4.2:** Target languages in the NIST LRE09 evaluation.

into account just top-50 eigenchannels, and trained by using the EM algorithm. To train these matrices, 350 hours were used for  $U_{CTS}$  (600 segments of approximately 150 seconds per the 14 languages used) and 550 hours for  $U_{VOA}$  (600 segments of around 150 seconds per the 22 available languages). This information is summarized in table 4.1.

## 4.2. Database Description

In order to perform experiments that can be compared with the reference system described above, the database used has been that provided by NIST LRE09 evaluation [NIST, 2009].

LRE09 database includes data coming from different audio sources: conversational telephone speech (CTS), used in previous evaluations, and broadcast data that contains telephone and non-telephone speech. That broadcast data consists of two corpora from past Voice of America (VOA) broadcast in multiple languages. One of these corpora (hereafter, VOA3) had VOA supplied language labels, but the labels of the other one (VOA2) were obtained by an automatic procedure. Furthermore, around 80 segments for each target language (of approximately 30 seconds duration each) had been audited for training purposes.

Regarding evaluation data, segments of 3, 10 and 30 second of duration from CTS and broadcast speech data are available to test the developed systems.

The languages considered in the mentioned evaluation are reported in table 4.2, but just those marked in blue have been used in the experiments of this work.

More details can be found in the NIST LRE09 evaluation plan [NIST, 2009].

### 4.3. Experimental Framework

As it was mentioned in section 3.2, some designing decisions should be made in order to select the structure that will be used for performing the experiments.

Some of these decisions remain fixed across all the experimental part of this work: convolutional DNNs have been the deep architecture selected to carry out the experiments and the stochastic gradient descent algorithm based on minibatches has been chosen as training algorithm.

Other fixed factor has been the windowing process with windows of duration of 10 milliseconds that has been applied to speech signals. Consistently, an input of 300 of length in the temporal dimension would correspond to a fragment of 3 seconds.

Two types of signal representation have been used as it was mentioned in section 3.2: SDCs and Mel-scale filter-bank outputs (hereafter, *MFBs*). In the case of *MFBs*, a normalization of zero mean and unit variance has been applied in order to help the learning process of the network and get a better performance. According to empirical results, when the network is fed with SDCs, this normalization does not seem to be needed. The reason might be the cepstral mean and variance normalization (CMN and CVN) applied when the cepstral representation based on SDCs is being obtained.

Furthermore, all the models used in the experiments presented in this Dissertation are tested on segments of 3 seconds of duration, in which just around 2 or 2.5 seconds are actual speech (the rest is supposed to be silence or noise). Since the implementation of the network requires a fix size of the input signal, the solution taken to the problem of variable sizes in the input has been to replicate the first part of the speech. Then, a right *padded* has been applied to obtain 3 seconds (or 300 frames) of speech as input signal.

The rest of the parameters involved in the network configuration (see *Configuration of the network*, section 3.2) are different for each experiment. Most of them use three convolutional (hidden) layers, with filter shapes of dimension 5 for the first two layers, and of dimension 2 for the last one. The shape of the max-pooling subsampling applied in the first two convolutional layers is, for most of the cases,  $2 \times 2$ , and, for the last layer, this shape is set to the dimension needed to get just a value ( $1 \times 1$  shape) at the end.

Regarding the partition of the datasets used to perform the experimental part, three different sets are required:

- Training set, composed of examples used to estimate the free parameters of the network (weights and bias).
- Validation set, formed by examples out of the training set used to perform model and configuration parameters selection and the “early-stopping” condition.
- Test set, examples used to evaluate the final generalization error and compare different algorithms in an unbiased way.

Language	# Files VOA2	# Files VOA3	# Total Hours
Amharic	379	-	16
Bosnian	5	866	36
Cantonese	782	-	33
Creole	56	2598	111
Croatian	80	373	19
Hausa	623	6237	286
Hindi	324	1464	75
Turkish	47	918	40
Ukrainian	606	738	56
Urdu	738	10842	483
Vietnamese	1872	-	78

**Table 4.3:** Amount of available data for each language used for development in the experiments performed. Each file corresponds with approximately 150 seconds of speech.

Depending on how the database was split, two types of experiments can be distinguished: 1) experiments where the development dataset of NIST LRE09 was divided into the three datasets needed; and 2) those experiments that used the development data of NIST LRE09 for training and validation sets and the evaluation data provided by NIST as test set.

In order to specify the source of the development data, table 4.3 shows the number of files that comes from *VOA2* and *VOA3* for each language used in the experiments.

Finally, as far as the type of experiments performed is concerned, they can be divided into three types:

- One vs. one (language pairs): just two languages are involved in this set of experiments. Then, the network has just two output units.
- All vs. all (closed-set): experiments with so many output units as languages are involved in the experiment (more than two languages). For this work, these experiments are constraint to six languages. Moreover, the same languages used to train are those that will be included in the test set (closed-set task).
- One vs. all: experiments where although six languages (in this work) are involved, just one of them is considered as target language.

In all cases, the output is a score (real value indicating a “likelihood measure”) for each unit, i.e. the network establishes a score of belonging to each language among those considered in the experiment performed. It should be taken into account that the predicted label is set to the class that has received the highest score (*argmax* function), and the Zero-One Loss depends on this decision.



Exp.	# Hidden layers	# Filters / layer	Filter shapes	Pool shapes
HU1	3	[12, 12, 12]	[(5, 5), (5, 5), (2, 2)]	[(2, 2), (2, 2), (1, 71)]
HU2	3	[20, 50, 30]	[(5, 5), (5, 5), (2, 2)]	[(2, 2), (2, 2), (1, 71)]
HU3	3	[6, 12, 15]	[(5, 5), (5, 5), (2, 2)]	[(2, 2), (2, 2), (1, 71)]
HU4	4	[6, 6, 6, 6]	[(5, 5), (5, 5), (5, 5), (11, 11)]	[(1, 2), (1, 2), (1, 2), (1, 24)]

**Table 4.4:** Configuration parameters for the experiments performed on the language pair Hindi - Ukrainian.

## 4.4. Results

In this section, the results that have been obtained during the experimental part of this work are detailed and analyzed. Results are presented by using DET (Detection Error Tradeoff) Curves, confusion matrices and error measures such as Zero-One Loss (ZOL), Equal Error Rate (EER) and an average cost (*meanCavg*) defined as in NIST [2009].

The remaining of the section is organized as follows. First, some experiments where just two languages are involved are presented (“one vs. one” models). Second, “all vs. all” models are performed; in these experiments, two sets of six languages each are used. Finally, some “one vs. all” experiments (just one target language) are shown.

### 4.4.1. “One vs. One” Experiments (*Language Pairs*)

This set of experiments includes those where just two languages are considered: one target language and one non-target language. Both training and test datasets are only composed of segments of speech from the two languages involved in the experiment.

#### ■ Hindi - Ukrainian (*HU*)

Four different experiments have been performed using these two languages. All of them use the same training, validation and test datasets (around 42 hours have been used for training, 18 hours for the validation set and 986 segments of 3 seconds of speech for testing). The input speech signal is represented by *MFBs*, each fragment corresponds with 3 seconds of speech and a normalization of zero mean and unit variance has been applied.

The models differ in the network configuration, as it is specified in table 4.4.

The test error results obtained in these four experiments are shown in table 4.5, where reference systems results are also included.

The ZOL percentage is worse in all cases comparing with the reference system that uses 1024 Gaussian components. That could be due to the simple decision that our system makes of taking as the predicted class that that achieves the maximum score.

However, according to the EER and meanCavg measures, HU1, HU2 and HU3 experiments outperformed the reference systems. It should be highlighted that HU4, which has four

Error Measure	Ref. System 512	Ref. System 1024	HU1	HU2	HU3	HU4
ZOL (%)	22.12	<b>17.90</b>	21.81	19.17	18.66	30.02
EER (%)	27.32	25.77	21.45	<b>19.30</b>	20.38	28.69
MeanCavg	26.52	22.77	21.25	<b>18.48</b>	19.64	27.74

*Table 4.5: Results experiment Hindi vs. Ukrainian.*

Error Measure	Ref. System 512	Ref. System 1024	TV1	TV2	TV3	TV4
ZOL (%)	21.47	<b>17.12</b>	25.56	20.26	26.53	23.63
EER (%)	27.21	21.69	24.03	<b>20.16</b>	26.74	23.26
MeanCavg	27.07	20.93	22.82	<b>18.48</b>	24.49	22.55

*Table 4.6: Results experiment Turkish vs. Vietnamese.*

hidden (convolutional) layers, achieves worse error rates than the other systems. Two possible reasons could be that the amount of available data is not enough to tune the parameters of the network or the selected configuration is not appropriated for this data.

- Turkish - Vietnamese (*TV*)

This set of experiments is similar to the previous one, but the aim is to distinguish between Turkish and Vietnamese languages. The four configurations of the network parameters are the same as the ones use in the Hindi - Ukrainian language pair. Regarding the datasets, the amount of data used is as follows: 40 hours for training, 17 hours for validation and 622 segments of around 3 seconds of speech for testing, approximately.

The results obtained for each system are summarized in table 4.6.

In this case, just the second model (TV2) outperforms the best reference system (1024 Gaussians) according to EER and meanCavg measures. The rest of the models are comparable but not better than the reference systems. A possible reason for this can be that almost all the files of Turkish and Vietnamese used to train the network come from the same part of the database (VOA2 in the case of Turkish and VOA3 in the case of Vietnamese). This can derive in an increase of the *overfitting* problem, and a worse generalization.

The small differences between EER and ZOL measures in the cases of DNN-based systems can be an indication of well alignment in the output scores (they are probabilities, scores normalized to sum one, which makes the range of output scores be small).

- Bosnian - Croatian (*BC*)

The last language pair of languages considered has been Bosnian and Croatian languages. These two languages are a challenging pair of languages due to they are pretty similar to each other.

Error Measure	Ref. System 512	Ref. System 1024	BC
ZOL (%)	39.26	<b>36.94</b>	46.58
EER (%)	48.40	46.28	<b>44.02</b>
MeanCavg	40.79	<b>38.42</b>	42.29

Table 4.7: Results experiment Bosnian vs. Croatian.

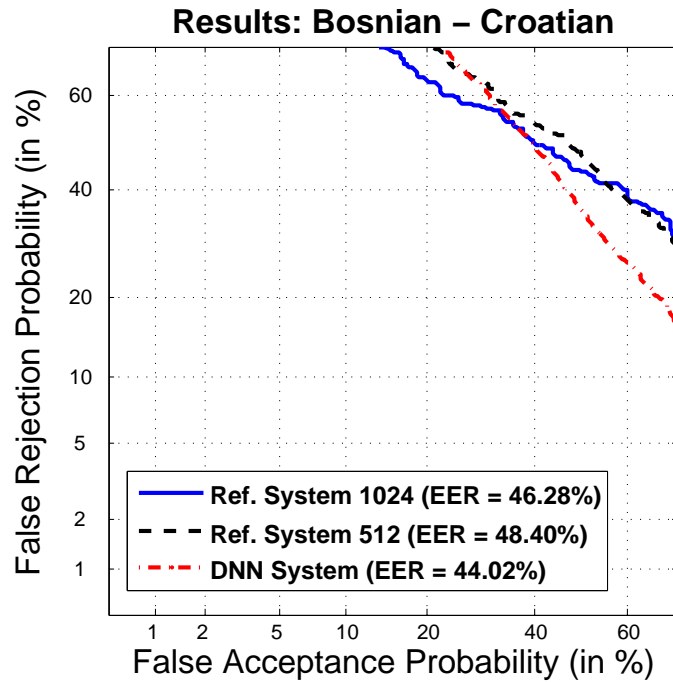
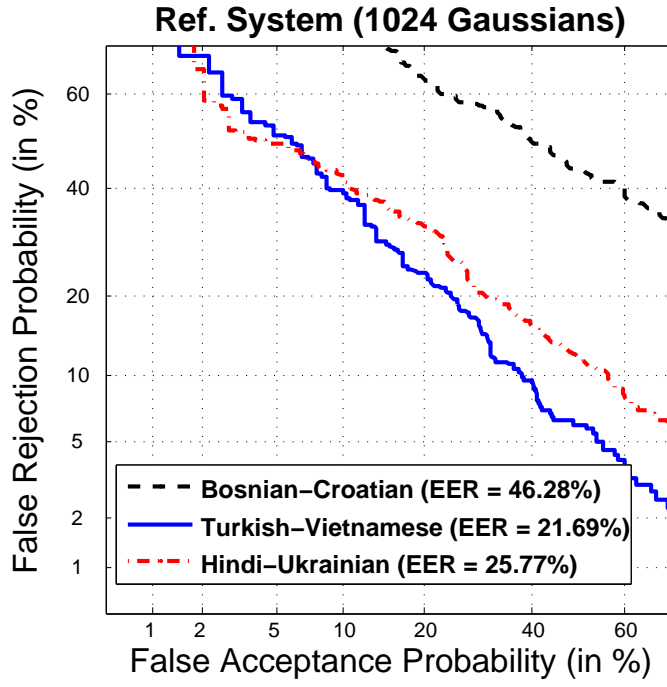


Figure 4.1: DET curves corresponding to results of the Bosnian vs Croatian experiment.

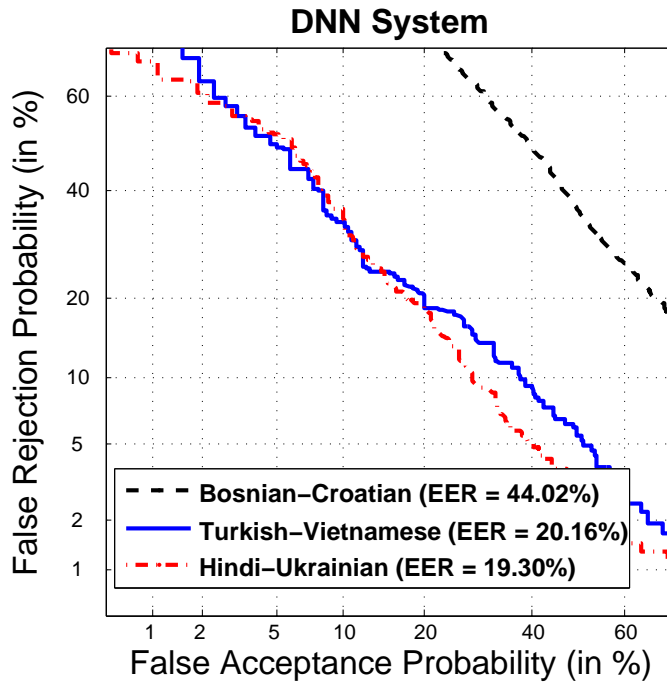
The configuration of the network has been the same that that used in the experiment HU1, and the amount of available data for this case has been around 21 hours for training, 9 hours for validation and 1022 segments of 3 seconds of speech for testing.

The results achieved by this model are shown in table 4.7. The EER is lower than that obtained by reference systems, as it can be seen in figure 4.1. The remaining error measures are worse than the reference systems ones.

Finally, figure 4.2 shows the DET curves of the reference system (with 1024 Gaussian components) (top) and the best (according to EER) DNN model (down) for each pair of languages involved in the experiments presented above.



(a) Reference system (1024 Gaussian components).



(b) Best DNN system (according to EER).

**Figure 4.2:** DET curves corresponding to pairs results of the reference system with 1024 Gaussian components (top) and the best DNN models according to the EER for each language pair (down).

#### 4.4.2. “All vs. All” Experiments (*Closed-set*)

Other set of experiments performed are those where more than two languages are involved. These experiments are based on a closed-set “all vs. all” model, where the same languages that are used to train the network are the expected ones in the testing stage.

Two different sets (of six languages each one) have been chosen. The first one uses a SDC representation of the input signal, as it was used in the reference system. The second is based on MFBs and a zero mean and unit variance normalization has been applied, as it was done in the “language pairs” experiments. More details and results are specified below.

##### 1. Experiments based on SDC representation

This set of experiments takes into account the six following languages: Amharic, Bosnian, Cantonese, Croatian, Hindi and Urdu. It should be highlighted that among this group of languages, very similar language pairs are included such as Bosnian-Croatian and Hindi-Urdu.

Two different experiments have been performed. Both of them use the same configuration of the network, specified in table 4.8.

The first experiment, identified as *ABCCHU-EVAL-TEST*, uses data from the development set provided by NIST LRE09, and splits them into training and validation datasets, while the test dataset is composed of data from the evaluation data of the database. Around 20 hours of speech are used for training, 5 hours for validation and 2469 segments of 3 seconds of speech for test have been used.

The second one (*ABCCHU-DEV-TEST*) uses just the development data from the NIST LRE09 database and splits them into the three datasets required by the algorithm. Then, approximately 15 hours of speech are used for training, 5 hours for validation and 5 hours for testing too (5940 segments of around 3 seconds of speech).

The second model described, as it can be seen in table 4.9, achieves better performance than the reference systems, but this does not happen with the first one. A possible reason could be that, in the second case, test examples are very similar to the training ones; indeed, they can be parts of the same utterances and share the same speakers. Thus, the network can be learning features that are not directly related with the language itself, so that *overfitting* can be occurring and a good generalization is not being achieved. The big *gap* between the validation and test error that exists in the first case could be an indication of this problem as well.

As it can be seen in the confusion matrices (tables 4.10, 4.11, 4.12 and 4.13), discriminating between Bosnian-Croatian and Hindi-Urdu segments of speech is one of the main error sources of the network, as it was expected due to the similarities that exist among this two language pairs. The last column shows the True Positive Rate (TPR) or Sensitivity for each language, calculated as  $\text{True Positives}/(\text{True Positives} + \text{False Negatives})$ .

Exp.	# Hidden layers	# Filters / layer	Filter shapes	Pool shapes
6 languages, SDCs	3	[12, 12, 12]	[(5, 5), (5, 5), (11, 11)]	[(2, 2), (2, 2), (1, 62)]

**Table 4.8:** Configuration parameters for the performed experiments based on SDC representation.

Error Measure	Ref. System 512	Ref. System 1024	ABCCHU-EVAL-TEST	ABCCHU-DEV-TEST
ZOL (%) (validation)	-	-	<b>38.87</b>	39.66
ZOL (%) (test)	51.15	47.43	60.96	<b>39.75</b>
MeanEER (%)	25.91	24.10	29.37	<b>20.20</b>
MeanCavg	25.36	23.36	28.69	<b>19.54</b>

**Table 4.9:** Results “all vs. all” experiments based on SDC representation (ABCCHU experiments).

		PREDICTED CLASS						TPR (%)
		Amharic	Bosnian	Cantonese	Croatian	Hindi	Urdu	
ACTUAL CLASS	Amharic	298	26	23	33	13	5	74.87
	Bosnian	31	142	26	129	19	8	40.00
	Cantonese	27	28	218	27	25	9	65.27
	Croatian	26	90	19	224	7	10	59.57
	Hindi	100	52	72	69	227	109	36.09
	Urdu	56	39	35	40	110	97	25.73

**Table 4.10:** Confusion matrix for the reference system (512 Gaussian components). Languages Amharic, Bosnian, Cantonese, Croatian, Hindi and Urdu.

		PREDICTED CLASS						TPR (%)
		Amharic	Bosnian	Cantonese	Croatian	Hindi	Urdu	
ACTUAL CLASS	Amharic	306	25	21	21	17	8	76.88
	Bosnian	29	163	16	116	14	17	45.92
	Cantonese	23	16	241	23	16	15	72.16
	Croatian	24	81	15	235	13	8	62.50
	Hindi	86	56	61	58	248	120	39.43
	Urdu	42	27	39	34	130	105	27.85

**Table 4.11:** Confusion matrix for the reference system (1024 Gaussian components). Languages Amharic, Bosnian, Cantonese, Croatian, Hindi and Urdu.

		PREDICTED CLASS						TPR (%)
		Amharic	Bosnian	Cantonese	Croatian	Hindi	Urdu	
ACTUAL CLASS	Amharic	250	8	35	49	24	32	62.81
	Bosnian	44	59	13	149	24	66	16.62
	Cantonese	48	18	168	16	21	63	50.30
	Croatian	56	53	20	168	24	55	44.68
	Hindi	119	8	39	68	146	249	23.21
	Urdu	47	5	18	43	49	146	47.40

*Table 4.12: Confusion matrix for the experiment ABCCHU-EVAL-TEST (test evaluation data).*

		PREDICTED CLASS						TPR (%)
		Amharic	Bosnian	Cantonese	Croatian	Hindi	Urdu	
ACTUAL CLASS	Amharic	663	36	85	61	66	65	67.93
	Bosnian	53	591	42	167	32	28	64.73
	Cantonese	68	23	699	18	55	41	77.32
	Croatian	55	125	26	643	64	53	66.56
	Hindi	95	35	64	88	447	210	47.45
	Urdu	102	53	69	98	246	331	36.82

*Table 4.13: Confusion matrix for the experiment ABCCHU-DEV-TEST (test part of the development dataset).*

Error Measure	Ref. System 512	Ref. System 1024	CHHTUV-300	CHHTUV-50
ZOL (%) (validation)	-	-	<b>26.4</b>	43.59
ZOL (%) (test)	46.05	<b>39.87</b>	60.91	67.78
MeanEER (%)	27.08	<b>22.86</b>	32.89	-
MeanCavg	26.11	<b>22.27</b>	31.96	-

**Table 4.14:** Results “all vs. all” experiments based on MFB representation (CHHTUV experiments).

## 2. Experiments based on MFB representation

The second set of “all vs. all” models are based on a MFB representation. In this case, segments of Creole, Hausa, Hindi, Turkish, Ukrainian and Vietnamese languages are used to train and test the network. The amount of available data has been increased in order to try to avoid the *overfitting* problem of the previous set of experiments. Then, 145 hours have been used for training, 62 hours for validation and 2 hours for testing, approximately.

Two experiments have been performed: one model have been trained on fragments of 300 frames in the time domain (3 seconds), and other, on segments of just 50 frames. Both have been tested on utterances of 3 seconds of duration, averaging the scores in the second case to obtain one score per utterance. It should be taken into account that in the second case, replicating the first frames, as it was done for the rest of the experiments, was not necessary.

The configuration parameters use the same values than in the *HU1* experiment.

Table 4.14 shows the validation and test error rates achieved by the two described models. Comparing with the reference system, both reach worse performance according to the test error. However, validation errors are pretty lower than test errors in both cases, which could be an indication that the problem of *overfitting* is happening.

Regarding the comparison between the two different models based on convolutional DNNs, it seems to be better the first one, which uses 300 frames of speech to train. This can make us supposed that the more information the network receives, the better generalization can be obtained.

Tables 4.15, 4.16 and 4.17 show the confusion matrices and the True Positive Rate (TPR) or Sensitivity for each language obtained by the different systems.

For the case of the experiment referred as *CHHTUV-300* (table 4.17), it can be observed that the worst classification results are obtained for Creole, Turkish and Vietnamese. These languages have most of their examples from the same part of the database (*VOA2* or *VOA3*) according to the information shown in table 4.3. This might cause less variance among the data used to train the network, and could be the reason for the *overfitting* observed in this model and for the bad generalization obtained.



		PREDICTED CLASS						TPR (%)
		Creole	Hausa	Hindi	Turkish	Ukrainian	Vietnamese	
ACTUAL CLASS	Creole	202	20	24	30	22	25	62.54
	Hausa	45	188	36	50	26	44	48.33
	Hindi	50	41	329	88	52	69	52.31
	Turkish	27	17	30	270	19	31	68.53
	Ukrainian	60	27	37	53	178	33	45.88
	Vietnamese	20	25	32	46	24	125	45.96

*Table 4.15: Confusion matrix for the reference system (512 Gaussian components). Languages: Creole, Hausa, Hindi, Turkish, Ukrainian and Vietnamese.*

		PREDICTED CLASS						TPR (%)
		Creole	Hausa	Hindi	Turkish	Ukrainian	Vietnamese	
ACTUAL CLASS	Creole	217	19	20	27	14	26	67.18
	Hausa	31	221	28	53	18	38	56.81
	Hindi	45	36	362	84	39	63	57.55
	Turkish	18	18	31	293	15	19	74.37
	Ukrainian	43	15	33	56	217	24	55.93
	Vietnamese	16	24	37	37	28	130	47.79

*Table 4.16: Confusion matrix for the reference system (1024 Gaussian components). Languages: Creole, Hausa, Hindi, Turkish, Ukrainian and Vietnamese.*

		PREDICTED CLASS						TPR (%)
		Creole	Hausa	Hindi	Turkish	Ukrainian	Vietnamese	
ACTUAL CLASS	Creole	66	19	55	12	128	29	21.36
	Hausa	30	122	73	14	98	23	33.89
	Hindi	17	57	331	31	150	27	54.00
	Turkish	31	23	124	72	82	32	19.78
	Ukrainian	9	6	41	7	257	53	68.90
	Vietnamese	15	71	67	11	52	42	16.28

*Table 4.17: Confusion matrix for the experiment CHHTUV-300, model of 300 frames.*

Target Language	EER (%)	
	Ref. System 1024	Conv. DNN Systems “one vs. all”
Amharic	<b>11.41</b>	16.22
Bosnian	<b>10.84</b>	16.21
Cantonese	17.62	<b>11.14</b>
Croatian	<b>12.80</b>	20.92
Hindi	30.96	<b>24.59</b>
Urdu	<b>27.92</b>	30.93

**Table 4.18:** Performance of reference system (1024 Gaussian components) (ATVS3, Gonzalez-Dominguez [2011]) and “one vs. all” models using convolutional DNNs on development dataset (per language).

#### 4.4.3. “One vs. All” Experiments

This third group of experiments consists of models in which just one language is the target language, and the rest of them (five languages in this case) are non-target.

The six languages involved are Amharic, Bosnian, Cantonese, Croatian, Hindi and Urdu. Six different “one vs. all” models have been developed, considering one of the six listed languages as target language. Then, each model will output two scores, one for each possibility: *being or not the target language*.

The configuration of the network is that shown in table 4.8, and each segment of speech is represented with 56 SDCs (the same configuration described for the reference systems).

For these experiments, development datasets provided by NIST LRE09 have been partitioned as follows: around 15 hours of speech for training, 5 hours for validation and 5 hours for testing. Approximately, the sixth part of each dataset belongs to the target language.

Table 4.18 shows the results of these “one vs. all” systems, comparing with the results obtained by the reference system (1024 Gaussian components). To be fair with the comparison, the results of the reference system that are included in this table are those achieved in a test set extracted from the development dataset, and not from the evaluation data provided by NIST LRE09.

As it can be seen in table 4.18, just the models for Cantonese and for Hindi as target languages outperform the reference system. All the models might be *overfitting* the data (as it was mentioned in section 4.4.2 for the experiment *ABCCHU-DEV-TEST*), since the test data are very similar to the training data. It might have occurred that training data of Cantonese and Hindi languages have had more variations and the model had not *overfit* as much as in the other cases.

## Chapter 5

# Conclusions and Future Work

THIS MASTER THESIS has been focused on exploring approaches based on deep neural networks in order to improve language recognition systems.

Thus, after a wide study of some techniques that have been applied to spoken language recognition (SLR) problems and other fields closely related to this one, the experimental part of this work has been presented. The aim of this part has been to develop a new system, based on acoustic features, by using convolutional DNNs as machine learning architecture.

Then, this chapter includes some conclusions that have been extracted during the development of this work, both from a theoretical and practical point of view. Some future research lines where this work could be applied will be described as well.

### 5.1. Conclusions

Chapter 1 has introduced basic concepts about SLR field and some ideas of what deep neural networks (DNNs) are and why they are a relatively new tool in the machine learning field. The motivation of this Dissertation, its main goals and its structure has been also included in the first chapter.

In Chapter 2, relevant systems and algorithms within the two fields in which this work is included (SLR and machine learning) have been briefly described, focusing on those topics that are more closely related to the main objectives of this Dissertation.

Applications of DNNs have been introduced in Chapter 3, explaining also the main ideas to develop a SLR system based on DNNs, which is the proposed method of this work.

Finally, the experimental part of this Dissertation has been described in Chapter 4, where the experiments performed and the results achieved have been detailed and analyzed.

The main conclusions that can be extracted from the theoretical and experimental parts of this work are presented below.

On the one hand, the main conclusions that want to be highlighted from a theoretical point of view are:

- DNNs can be considered a *breakthrough* within the machine learning field, due mostly to their advantages over shallow neural networks (those with just one hidden layer). Among them, it can be highlighted the following ones:
  - Capability to model complex functions when the amount of available data is enough to estimate their free parameters.
  - Automated discovery of abstract representation of the data, overcoming shallow architectures dependency on *human-crafted* features.
  - Shallow architectures, as *local* estimators, are limited when a high variant function wants to be represented, i.e., variations of the input space is a non-linear manifold. This results in a poor *non-local* generalization when variations not seen in the training set appear in the test set. In this sense, it is needed a number of examples proportional to the number of variations to be covered. This limitation can be overcome by using a deep architecture (*non-local* estimator), which introduces non-linear transformations and allows a more *compact* representation of the function that wants to be modeled.
- There are numerous deep architectures and algorithms and, depending on their properties, could be suitable for different applications. Moreover, lots of different configurations of the same networks can be used to develop different models and, according to research works on this field, it seems that the best model for solving a certain problem can not be selected in a generic way, since the configuration and the architecture itself is highly dependent on data.
- The recent success of DNNs applied to speech recognition tasks encourages the use of these architectures on other related fields, such as SLR. They seem to model adequately the speech signal, which can help in the SLR task.
- Convolutional DNNs can be considered a good starting point due to their success in different computer vision and speech processing tasks. They are “easier” to train than other deep structures, and their specific properties reduce the number of free parameters, which means they need less data and lower computational capacity to reach good results.

On the other hand, some conclusions from the experimental part of this work can be extracted as well:

- Results achieved by the developed system of SLR by using convolutional DNNs are not always better than those that the reference system obtains. However, they are at least comparable, which seems to be a good starting point. Moreover, a fusion between both systems might outperform the individual systems, since they can be considered uncorrelated approaches: *discriminative versus generative* models.
- The new DNN-based system could be considered as a *baseline* system and the starting point to get improvements in the SLR task by following this approach.

- Comparing with the performance of the reference systems, the new developed systems seem not to generalize as well as it should be. When test data are quite different than training data, test error rates are markedly increased. One possible reason could be the *overfitting* problem, since a big *gap* between test and validation error exists. Also, available data are not enough different among them to collect all the variability sources that can appear in speech signals, which can cause this not good generalization as well.
- The huge amount of possible configuration of the networks can be considered a disadvantage of this kind of architectures and can make that the selected configuration of the model for a certain problem could not be the best option for it.
- The big memory and computational requirements of these training algorithms can be mentioned as drawbacks. In this case, some experiments have needed around three days to estimate “adequately” the network parameters (*weights and bias*) running in a CPU with 24 cores.

## 5.2. Future Work

A number of research lines arise from the work carried out in this Master Thesis. Among them, following ones can be highlighted:

- **Exploring DNNs configurations:** following the same approach carried out in this work, different configuration of convolutional DNNs will be tried in order to get better performance of the system.
- **Adding a *back-end* module:** in the experimental part of this work, the predicted class has been selected just taking that corresponding to the maximum score. More complex modules and techniques will be developed so that it can be taken advantage of the underlying information of the output scores of the network.
- **Exploring different types of deep architectures:** using other kind of structures, such as Restricted Boltzmann Machines (RBMs) or Deep Belief Networks (DBNs), will let us use *unsupervised* learning algorithms. This could be a key for improving SLR systems performance.
- **Synthetic data:** in order to avoid the *overfitting* problem, more data can be generated synthetically so that the new examples could collect different variability sources. Also, some techniques such as “*dropout*” [Hinton *et al.*, 2012b] will be implemented and analyzed to see how they affect in the SLR experiments performed.
- **Exploring other related research lines:** this work might be extended to other research fields such as, for example, speaker recognition area. Possible starting points can be the use of DNNs to obtain a new representation of the input signal or to develop a complete system following the work of Stafylakis *et al.* [2012].



## Appendix A

### Short Biography

Alicia Lozano-Diez was born in 1989. She started her Computer Science Engineering and Mathematics studies in 2007 and received the MSc degree in Computer Science Engineering and the MSc degree in Mathematics in 2012 from Universidad Autonoma de Madrid, Spain. Since June of 2012, she is with the Biometric Recognition Group - ATVS at the Universidad Autonoma de Madrid, where she is currently collaborating as an assistant researcher pursuing the PhD degree. Her research interests include speech signal processing and pattern recognition. Her current research focuses on language recognition by using deep neural networks.





# References

- E. Ambikairajah, H. Li, L. Wang, B. Yin, and V. Sethu. Language identification: A tutorial. *Circuits and Systems Magazine, IEEE*, 11(2):82–108, 2011. ISSN 1531-636X. 3, 9
- Y. Bengio. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2(1):1–127, 2009. Also published as a book. Now Publishers, 2009. XIII, 3, 12, 13, 15, 19
- Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. Greedy layer-wise training of deep networks. pages 153–160, 2007. URL <http://www.iro.umontreal.ca/~lisa/pointeurs/BengioNips2006A11.pdf>. 3, 17
- C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 1st ed. 2006. corr. 2nd printing 2011 edition, Oct. 2007. ISBN 0387310738. URL <http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0387310738>. 17
- W. Campbell, J. Campbell, D. Reynolds, D. Jones, and T. Leek. High-level speaker verification with support vector machines. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, volume 1, pages I-73–6 vol.1, 2004. 11
- D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber. Deep big simple neural nets excel on handwritten digit recognition. *CoRR*, abs/1003.0358, 2010. XIII, 21, 22
- J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR09*, 2009. 21
- L. Deng and X. Li. Machine learning paradigms for speech recognition: An overview. *IEEE Transactions on Audio, Speech & Language Processing*, 21(5):1060–1089, 2013. 1, 22
- D. Erhan, P.-A. Manzagol, Y. Bengio, S. Bengio, and P. Vincent. The difficulty of training deep architectures and the effect of unsupervised pre-training. pages 153–160, Apr. 2009. 3, 17
- J. Gonzalez-Dominguez. *Session Variability Compensation in Automatic Speaker and Language Recognition*. PhD thesis, Universidad Autonoma de Madrid, November 2011. xvi, 42
- J. Gonzalez-Dominguez, I. Lopez-Moreno, J. Franco-Pedroso, D. Ramos, D. T. Toledano, and J. Gonzalez-Rodriguez. Multilevel and session variability compensated language recognition: Atvs-uam systems at nist ire 2009. *IEEE Journal on Selected Topics in Signal Processing*, 2010. article in press. XIII, xv, 3, 11, 29, 30
- G. Hinton, L. Deng, D. Yu, G. Dahl, A. rahman Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition. *Signal Processing Magazine*, 2012a. XIII, 4, 13, 18, 22, 23, 24
- G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural Comput.*, 18(7):1527–1554, July 2006. ISSN 0899-7667. URL <http://dx.doi.org/10.1162/neco.2006.18.7.1527>. 3, 14, 17, 18

- G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580, 2012b. 45
- N. Jaitly, P. Nguyen, A. Senior, and V. Vanhoucke. Application of pretrained deep neural networks to large vocabulary speech recognition. In *Proceedings of Interspeech 2012*, 2012. 4
- P. Kenny, G. Boulianne, and P. Dumouchel. Eigenvoice modeling with sparse training data. *Speech and Audio Processing, IEEE Transactions on*, 13(3):345–354, 2005. ISSN 1063-6676. 29
- A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In P. Bartlett, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1106–1114. 2012. URL [http://books.nips.cc/papers/files/nips25/NIPS2012\\_0534.pdf](http://books.nips.cc/papers/files/nips25/NIPS2012_0534.pdf). XIII, 21, 23
- Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Intelligent Signal Processing*, pages 306–351. IEEE Press, 2001. 12, 19, 21, 25
- Y. Lecun and C. Cortes. The mnist database of handwritten digits. URL <http://yann.lecun.com/exdb/mnist/>. 21
- H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *ICML*, page 77, 2009a. 24, 25
- H. Lee, Y. Largman, P. Pham, 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. 2009b. XIII, 4, 22, 24, 25
- LISA. *Deep Learning Tutorial*. University of Montreal, <http://deeplearning.net/tutorial/>. 16, 27
- A.-r. Mohamed, G. E. Dahl, and G. Hinton. Acoustic modeling using deep belief networks. *IEEE Trans. on Audio, Speech and Language processing*. URL [http://www.cs.toronto.edu/~hinton/absps/speechDBN\\_jrnl.pdf](http://www.cs.toronto.edu/~hinton/absps/speechDBN_jrnl.pdf). 4, 13, 22, 23, 25
- T. K. Moon. The expectation-maximization algorithm. *IEEE Signal Processing Magazine*, 13(6):47–60, Nov. 1996. ISSN 10535888. URL <http://dx.doi.org/10.1109/79.543975>. 23
- NIST. The 2009 nist language recognition evaluation plan. 2009. URL [http://www.itl.nist.gov/iad/mig/tests/lre/2009/LRE09\\_EvalPlan\\_v6.pdf](http://www.itl.nist.gov/iad/mig/tests/lre/2009/LRE09_EvalPlan_v6.pdf). 30, 33
- H. Poon and P. Domingos. Sum-product networks: A new deep architecture. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 689–690, 2011. 4
- M. Ranzato, F. J. Huang, Y.-L. Boureau, and Y. Lecun. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007. 19
- R. Salakhutdinov and G. E. Hinton. Deep boltzmann machines. *Journal of Machine Learning Research - Proceedings Track*, 5:448–455, 2009. 4
- T. Serre, G. Kreiman, M. Kouh, C. Cadieu, U. Knoblich, and T. Poggio. A quantitative theory of immediate visual recognition. In *Progress in Brain Research, Computational Neuroscience: Theoretical Insights into Brain Function*, volume 165, pages 33–56. 2007. URL [http://web.mit.edu/serre/www/publications/Serre\\_etal\\_PBR07.pdf](http://web.mit.edu/serre/www/publications/Serre_etal_PBR07.pdf). 12

- T. Stafylakis, P. Kenny, M. Senoussaoui, and P. Dumouchel. Plda using gaussian restricted boltzmann machines with application to speaker verification. In *INTERSPEECH*. ISCA, 2012. 45
- P. A. Torres-carrasquillo, E. Singer, M. A. Kohler, and J. R. Deller. Approaches to language identification using gaussian mixture models and shifted delta cepstral features. In *Proc. ICSLP 2002*, pages 89–92, 2002. XIII, 8, 9
- R. Vogt and S. Sridharan. Explicit modelling of session variability for speaker verification. *Computer Speech & Language*, 22(1):17–38, 2008. 29
- M. Zissman and E. Singer. Automatic language identification of telephone speech messages using phoneme recognition and n-gram modeling. In *Acoustics, Speech, and Signal Processing, 1994. ICASSP-94., 1994 IEEE International Conference on*, volume i, pages I/305–I/308 vol.1, 1994. 10