Federal University of Rio Grande do Norte
Center for Exact and Earth Sciences
Bachelor in Computer Science

An Investigation of Type-1 Adaptive Neural Fuzzy Inference System for Speech Reconigtion

Thales Aguiar de Lima

Natal-RN
June 2018

Thales Aguiar de Lima

An Investigation of Type-1 Adaptive Neural Fuzzy Inference System for Speech Reconigtion

Undergraduate Report presented as a partial requirement for obtaining the degree of Bachelor in Computer Science.

Advisor

Márjory da Costa Abreu, PhD

FEDERAL UNIVERSITY OF RIO GRANDE DO NORTE – UFRN

Natal-RN

June 2018

Universidade Federal do Rio Grande do Norte - UFRN
Sistema de Bibliotecas - SISBI
Catalogação de Publicação na Fonte. UFRN - Biblioteca Setorial Prof. Ronaldo Xavier de Arruda - CCET

Lima, Thales Aguiar de.
An investigation of type-1 adaptive neural fuzzy inference system for speech reconigtion / Thales Aguiar de Lima. - 2018.
45f.: il.

Monografia (graduação) - Universidade Federal do Rio Grande do Norte, Centro de Ciências Exatas e da Terra, Departamento de Informática e Matemática Aplicada, Bacharelado em Ciência da Computação. Natal, 2018.
Orientadora: Márjory da Costa Abreu.

1. Computação. 2. ANFIS. 3. Speech recognition. 4. Phoneme. 5. Fuzzy sets. 6. Neural networks. I. Abreu, Márjory da Costa. II. Título.

RN/UF/CCET CDU 004

Elaborado por Joseneide Ferreira Dantas - CRB-15/324

Undergraduate thesis under the title *An Investigation of Type-1 Adaptive Neural Fuzzy Inference System for Speech Reconigtion* presented by Thales Aguiar de Lima and accepted by the Department of Informatics and Applied Math of the Center for Exact and Earth Sciences of the Federal University of Rio Grande do Norte, being approved by all the members of the examining board specified below:

PhD. Márjory da Costa Abreu
Advisor
Department of Informatics and Applid Mathmatics
Federal University of Rio Grande do Norte

PhD. Laura Emmanuella Alves dos Santos de Oliveira
Agricultural School of Jundiaí
Federal university of Rio Grande do Norte

PhD. Plácido Antônio de Souza Neto
Academic Direction of Management and Information Technology
Federal Institute of Rio Grande do Norte

Natal-RN, June 19 of 2018.

To everyone who helped me in this journey.

Acknowledgments

First, I want to thank my family for all the support without which I could not accomplish this work. Besides, I also want to thank Rita for being my psychologist, always taking care of me and believing in me more than I do.

Next, I want to thank Márjory da Costa, my advisor, for the great guidance and patience along the journey to accomplish this work.

Finally, I want to thank all my friends who helped me until this point. Then, "so long, and thanks for all the fish" - Adams, Douglas.

“Too much pain, not enough profit.”

Viddar, The Collector - Diablo 3

An Investigation of Type-1 Adaptive Neural Fuzzy Inference System for Speech Reconigtion

Author: Thales Aguiar de Lima

Advisor: Márjory Da Costa Abreu, PhD

Abstract

Using voice for user recognition is something that humans do since the beginning and it a very natural ability. Being able to recognise the user by its voice is very important, but, in some cases, being able to recognise what is being said automatically can have very interesting and useful security applications. Thus, speech recognition has been experiencing a increasingly growth in attention in the last years, following the advancements of the machine learning field. Since this is a very complex problem and can have interference from several different sources, there has been widely different approaches to perform this task, very often with high cost, and more frequent than not, with results that are dependent on the high quality of the data, which is not always the case. In this paper we present an Type-1 Adaptive Neural Fuzzy Inference System for speech recognition on MOCHA-TIMIT repository. Besides, we also used the *Mel-Frequency Cepstrum Cofficient* and *Filter-Banks* feature extraction methods aiming to translate speech to text with low or medium quality samples and still have a good results when dealing with speech recognition.

Keywords: ANFIS, Speech recognition, Phoneme, Fuzzy sets, Neural Networks.

Contents

List of figures

List of tables

List of abbreviations

ANFIS – Adaptive Network Fuzzy Inference System

ASR – Automatic Speech Recognition

CNN – Convolutional Neural Networks

DBN – Deep Belief Network

DCT – Discrete Cosine Transform

DFNN – Deep Fuzzy Neural Network

DNN – Deep Neural Networks

DRNN – Deep Recurrent Neural Network

FNN – Regular Fuzzy Neural Networks

GRBM – Gaussian Restricted Boltzmann Machine

GMM – Gaussian Mixture Models

HMM – Hidden Markov Models

KLD – Kullback-Leibler Divergence

LDC – Linguistic Data Consortium

LVSR – Large Vocabulary Speech Reconigtion

LSE – Least Square Estimation

MFCC – Mel-Frequency Cepstrum Coefficient

ML – Maximum-Likelihood

MMI – Maximum Mutual Information

MPE – Minimum Phone Error

MSE – Mean Squared Error

NN – Neural Network

PER – Phone Error Rate

PLA – Piecewise Linear Approximation

RBM – Restricted Boltzmann Machine

WER – Word Error Rate

1 Introduction

Speech recognition is a topic that has been studied since the 1960s, where it was possible to process a small set of words and numbers from audio samples (JUANG; RABINER, 2005). The advancements of computational resources capacity and researches on the artificial intelligence field has provided a large progress on speech recognition research (AREL; ROSE; KARNOWSKI, 2010). Hence, it is possible to find examples from the main information technology companies, like Alexa[1] from Amazon and Siri[2] from Apple that are specific for speech recognition. Besides, speech recognition has a large application field, as for instance, the digital inclusion of blind people (RAN; HELAL; MOORE, 2004).

There is also the fact that as a result of the computation advancements and information inclusion the amount of data is increasing in a high rate (MANJUNATH; HEGADI; RAVIKUMAR, 2010). Such data contains various information that could be used by many applications, such as *cookies* on the internet, for example, where a set of user data is collected and something is inferred by analysing it (SRIVASTAVA et al., 2000). However, the huge amount of different types of multimedia like videos, audios, speech and text with their natural unstructured form make the manual extraction of useful information a complicated task. Therefore, the support of tools to extract this information can help to cope with this difficulty. Speech recognition techniques can be applied to this problem by structuring the information and creating a set of data which can then be easily mined (MANJUNATH; HEGADI; RAVIKUMAR, 2010).

With the enhancement of the extraction techniques, we also have an increasingly attention for the information security. As defined by (VENTER; ELOFF, 2003) information security is the protection of information and minimising the risk of exposing information to unauthorised parties. The high accuracy and the different types of data collected from users is leading to complex inferences about the users behaviour (MANJUNATH; HEGADI; RAVIKUMAR, 2010). In this scenario, speech recognition also plays an important

[1] https://www.alexa.com/

[2] https://www.apple.com/ios/siri/

role by translating the voice of the user, allowing the creation of more complex types of identification processes as well as the speaker identification that can add even more complexity to the user data protection by analysing the frequency of its voice. Those tasks can help to protected the user data from unauthorised access, as only an user with a relatively close voice frequency and in knowledge of the identification process could be able to access these information (HASAN et al., 2004).

It is important to note that speech recognition has different meanings. The two main interpretations are either related with speaker identification (MACMILLIAN, 1986) or with a process that identify what is being said (RABINER; JUANG, 1993). In this work, speech recognition refers to the transformation of an utterance into the text representation of each speech sound, e.g. Phoneme.

Since the early investigations on the speech recognition field, Hybrid Systems had better results than traditional methods (YU; DENG; DAHL, 2010). The combination of different techniques is still an important topic on the field, although recent works on neural networks or deep learning are slowly overcoming the outcomes from hybrid systems. In this context, the use of neural networks or deep neural networks is providing good outcomes for processing voice to text, which can be seen in the most recent works that are using these approaches, such as in (DAHL et al., 2012) where it was common to use a hybrid of a Gaussian Mixture Models (GMM) and Hidden Markov Models (HMM).

Therefore, this work propose an investigation of Adaptive Fuzzy Inference System (ANFIS) for Automatic Speech Recognition (ASR). This is a class of adaptive networks, proposed by (JANG; SUN; MIZUTANI, 1997), that are equivalent to Fuzzy Inference Systems. As defined in (COPPIN, 2004) a fuzzy inference system, or fuzzy expert system, is an expert system which relies on a set of rules given by an expert (external), then a number of crisp values are fed to the system and the rules are used to generate a recommendation. The ANFIS was proposed by (JANG, 1993) and it consists on a hybrid system that combines the learning capacities of Neural Networks with the Fuzzy Logic ability to deal with vagueness information.

This approach has not been in-depth studied, and only few works like this exist in the field of speech recognition. Thus, our main goal is to investigate further the effectiveness of such approach. The main steps to reach the classification for this work are to (1) collect the data, (2) extract the features, (3) execute a feature cleaning and (4) classify the data. Each of these steps can be represented by an technique, for instance in (AVCI; AKPOLAT, 2006) the steps (2) and (4) are respectively performed by Wavelet Packet Decomposition and a Type-3 ANFIS (Takagi and Sugeno's model), while in this work they are performed by a Mel Frequency Cepstrum Coefficient and a Type-1 ANFIS (Tsukamoto model).

1.1 Work organization

This work is organized as follows, Chapter 2 describes the main databases used in the works from the last 5 years, just as the reasons to chose the database used in this work. In Section 2.2 the steps for the feature extraction are explained. A brief discussion about the related classifiers and their results, as well as an explanation upon the classifier used in this work is shown in Chapter 3. The Chapter 4 presents a detailed description of the steps to structure the model and how to find the parameters that best fit it using the Least Square Estimation, thus explaining the learning procedure for the Hybrid System of this work. Finally, we finish with the conclusion in Chapter 6 presenting future work goals.

2 Speech recognition: database analysis and feature extraction techniques

Since our main goal is to analyse the predictability of Neural Fuzzy structures for speech recognition, we had to find the ideal database. There are several database/corpus available that can be used for speech recognition analysis. From the investigated databases in this work, their audio samples can be categorised as clean or noisy and were collected from different places, applications and environments. The following section will briefly describe the corpus that are frequently used in the literature and we finish the section with the description of the database that will be used in this work.

2.1 Databases Description

The TIMIT Corpus is available in the Linguistic Data Consortium (LDC) catalogue. It is one of the oldest corpus currently available and has a total of 630 speakers of different American dialects, with around 10 phonetically rich sentences resulting in approximately 6300 utterances sampled at 16-bit and 16kHz. Since this is the oldest database, it is, so far the most used, being explored in works such as (ABDEL-HAMID et al., 2012), (GRAVES; MOHAMED; HINTON, 2013) and (ABDEL-HAMID; DENG; YU, 2013). The TIMIT corpus is not a freely used database, hence it is not coherent with the resources of this work.

The second most used corpus is the Bing Voice Search corpus. This database consists of utterances collected from the Bing Voice Search app, thus the audio samples have a wide variety of noise: multiple voices, cars sounds, music, etc. It is mostly used for large vocabulary speech recognition (DAHL et al., 2012) and for tests of noise removal (ABDEL-HAMID; DENG; YU, 2013). The utterances of this corpus are mostly short messages, often some instruction given by the user , all sampled at 8kHz, with approximately 40 hours of audio and 52 thousand utterances. Since Bing is a property of Microsoft, this set is not publicly available and therefore is not used for this research.

The third corpus that can be found in the literature is Aurora 4, which is partially publicly online and can be downloaded at (HIRSCH; PEARCE, 1997). This set is basically the Wall Street Journal (WSJ0) and TIDigits corpus with noise (SELTZER; YU; WANG, 2013) added by a program that is described at their download page. Even though this is a publicly database, it was not used in this work since our goal is to verify the applicability of acoustic model with Type-1 ANFIS for phoneme or speech recognition, and, for simplicity, we decided not to handle noise and therefore the chosen database must be of clean audio utterances.

Other corpus recently collected were the Switchboard (YU et al., 2013), XBOX voice search, Android and YouTube Voice Search (JAITLY et al., 2012). These sets have a real-world vocabulary with noise audio samples, except Switchboard but it does not have information to extract the phonemes. Furthermore, they are also not available for public use since most of them are from Google or Microsoft. Thus, these databases were not used in this research.

The CMU AN4 (GROUP, 1991) database, is mostly composed of alphanumeric audios recorded in Carnegie Mellon University at 1991. The goal of this database was to be used in the work presented in (ACERO, 1990). The subjects were asked to pronounce numbers, random phrases, and a fake personal identification, although some may be real. Therefore this database has 948 training and 130 test utterances at 16kHz and 16 bit presenting phone number, addresses, names, etc. The training set has 74 different speakers where 21 are female and 53 are male voices, while the test set has 10 different speakers, 3 female and 7 male voices. The CMU ARTIC database was also recorded at Carnegie Mellon University as CMU AN4. This corpus consists into a single male and a single female speaker, both of them experienced voice talent with 1132 prompts.

For this work, we have chosen to use the MOCHA-TIMIT corpus. This database is composed of 460 short sentences in Southern English language. Currently it has two speakers (one male and one female), each of them had an audio recorded for every sentence. Thus, this dataset has a total of 920 utterances recorded at 16kHz and 16bit. Even though the most popular databases are bigger than MOCHA-TIMIT, the latter has a file with time periods and the labels spoken in this period. Therefore, MOCHA-TIMIT would be better to work with phoneme recognition since other would need a pre-processing to create a labeled phoneme period of time. Although, after extracting the phonemes from MOCHA-TIMIT, it resulted in 12.454 utterances and 39 phonemes for the each speaker,

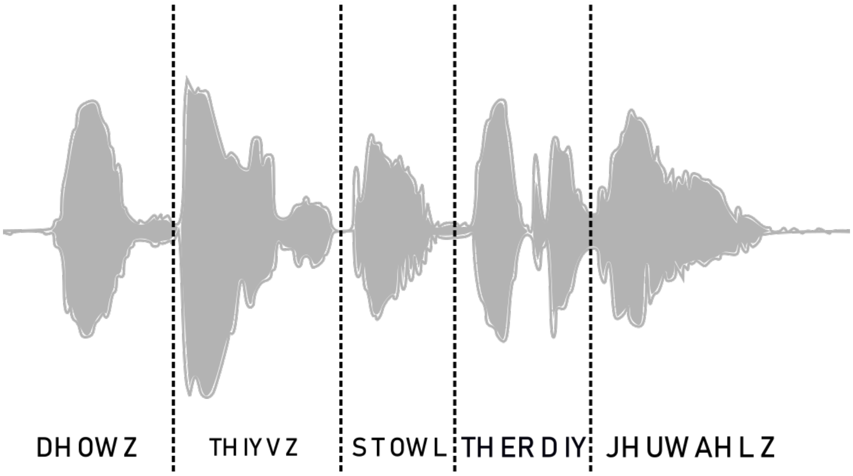


Figure 1: Waveform of `msak0_003` prompt for the phrase "Those thieves stole thirty jewels". Each range separate the words, which are phonetically represented.

thus resulting into a total of 24.408. Table 1 shows some samples from this database, as well as their respective phonetic translation. Figure 1 shows a waveform of an utterance from this database.

Table 1: Several prompts from MOCHA-TIMIT database

Prompt phrase	Phonemes
Is this seesaw safe?	IH Z . DH IH S . S IY S AO . S EY F
This was easy for us	DH IH S . W AA Z . IY Z IY . F AO R . AH S
She is thinner than I am	SH IY . IH Z . TH IH N ER . DH AE N . AY . AE M .
He will allow a rare lie	HH IY . W IH L . AH L AW . AH . R EH R . L AY .
I took her word for it	AY . T UH K . HH ER . W ER D . F AO R . IH T .
Is she going with you?	IH Z . SH IY . G OW IH NG . W IH DH . Y UW .

This section described the most popular databases used for speech recognition, showing their main characteristics, the motivation to use the selected database, why the other were discarded and the shape of the utterances in this work. The following section will explain the steps for the extraction of features from the dataset and how the features vector is composed.

2.2 Feature Extraction

As already mentioned, since we are investigating the efficiency of using a an adaptive neural fuzzy inference system for speech recognition, it is of fundamental importance to carefully understand and select the most appropriate feature selection technique. In this section, an introduction to the feature extraction technique used in this work is shown. Thus, we are going to describe the construction of the feature vectors used for the tests in this work.

As the data is composed of basically audio waves, it is necessary to transform the input into a data type that the neural network can manipulate. In this sense, a vector $v = (m_1, \ldots, m_k)$ is built, where k is the number of features which stands for the number of cepstrums. It is important to note that the order of the features are very relevant, for example, if a given input of a speaker that says "Hello" is then recognised as "ollHe", it is not correct even though the letters were precisely identified. Therefore, some works have already investigated the use of word alignment(ABDEL-HAMID et al., 2014)(ABDEL-HAMID et al., 2012) to sort the results.

In (HINTON et al., 2012a) the authors show some of the most relevant extractors that can be found in the literature. The *Mel-Frequency Cepstral Coefficient* (MFCC) is proposed as an alternative to *Filter-Banks Coefficients* because they are less strongly correlated. Even though, in Convolutional Neural Networks (CNN) the convolutional layer is not well applicable to MFCC because of the Discrete Cosine Transform (DCT) based uncorrelation transform, thus the filter-bank features, linear spectrum, or the Mel-Scale spectrum are good choices for this sort of Neural Network (NN) (ABDEL-HAMID et al., 2012). For example, in (GRAVES; MOHAMED; HINTON, 2013) and (ABDEL-HAMID; DENG; YU, 2013) the authors used feature extractor based on filter-banks, while in (KIM; STERN, 2012) the MFCC was used and (SELTZER; YU; WANG, 2013) investigated the results of hist network with both extraction methods. In (HINTON et al., 2012a), it is visible that the MFCC had a better performance for the TIMIT corpus, with a 1.7% better Word Error Rate (WER).

The Mel Scale attempts to simulate a human ear perception spectrum that is calculated for every given frequency with the Equation 2.1, and it is more sensible to high frequencies where the perception to different noises is more negligible. The MFCC is obtained by mapping the Fourier Transform of a window, as the divisions in Figure 1 but in a smaller interval of the frequency to the Mel Scale. After mapping the frequencies, the logs and the discrete cosine transform are applied to result in the MFCC.

$$m = 2595 log \left(1 + \frac{f}{700} \right) \tag{2.1}$$

To extract the features using MFCC firstly, a pre-emphasis is applied to amplify the higher frequencies. This stage will prevent numerical issues for the Fourier transform, improve the noise robustness of the acoustic model, and balance the frequency spectrum. The

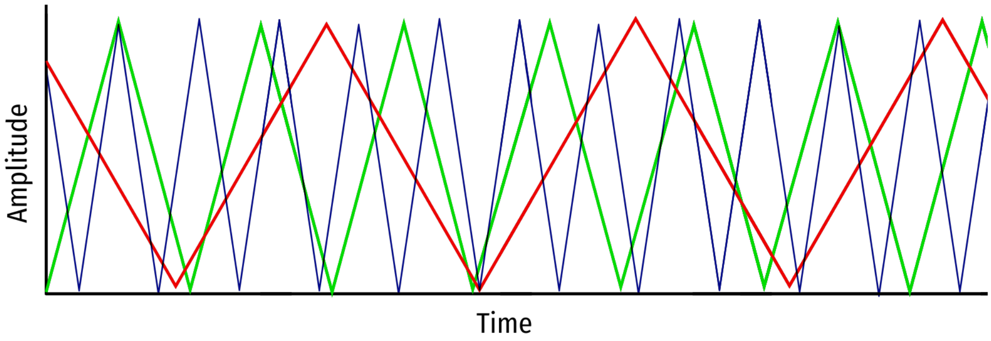


Figure 2: Example of triangular wave along time (horizontal) and amplitude (vertical) axis.

pre-emphasis can be done by applying $w^{'}(t) = w(t) - \alpha w(t-1)$ on a wave w and time t for a given $0 \leq \alpha \leq 1$, resulting on a new wave $w^{'}$. Then the resulting sound wave must be divided into short periods of time, small enough to consider that the wave has no change on shape, also known as frames and an windowing function is applied to them with the objective to smooth the curves. There are several methods for creating windows, but the most common is the Hamming Window (BLACKMAN; TUKEY, 1958) which stands for the equation $w[n] = 0.54 - 0.46 \cos\left({}^{2\pi n}/_{N-1}\right)$ where $0 \leq n \leq N$ and N is number of frames. After that, we can apply the Fourier transform on each frame to calculate the distribution of power over the frequency components, that is the power spectrum of them. Finally, the filter banks are computed, by applying triangular frequencies on mel scale. At this point the result is a set of power-levels varying with time for every cepstrum. So we can apply an normalisation to each of them and transform the metric to the mel scale producing each element of v, which is composed of the cepstrum of each frame. Due to the goal of this paper is to verify the outcomes of a Adaptive Network Fuzzy Inference System, the signal processing will be done with a third-part software, or algorithm. For instance, this work will be using the extractor made by Haytham Fayek [1].

Figure 3 shows an example of how the features are extracted from the utterances. In the first step, the portion of the utterance that represents a phoneme is extracted, then the pre-emphasis and framing is applied to the wave, followed by the window function. The second step consists on applying the Fourier-Transform to each frame and then the MFCC to obtain a matrix with the features. Posteriorly, a normalisation is applied to T in order to obtain the feature vector v, where $m_i = \sum_1^j n_{ji}/j$.

This section presented the feature extraction and described how each step is done,

[1] https://goo.gl/cBnx5a

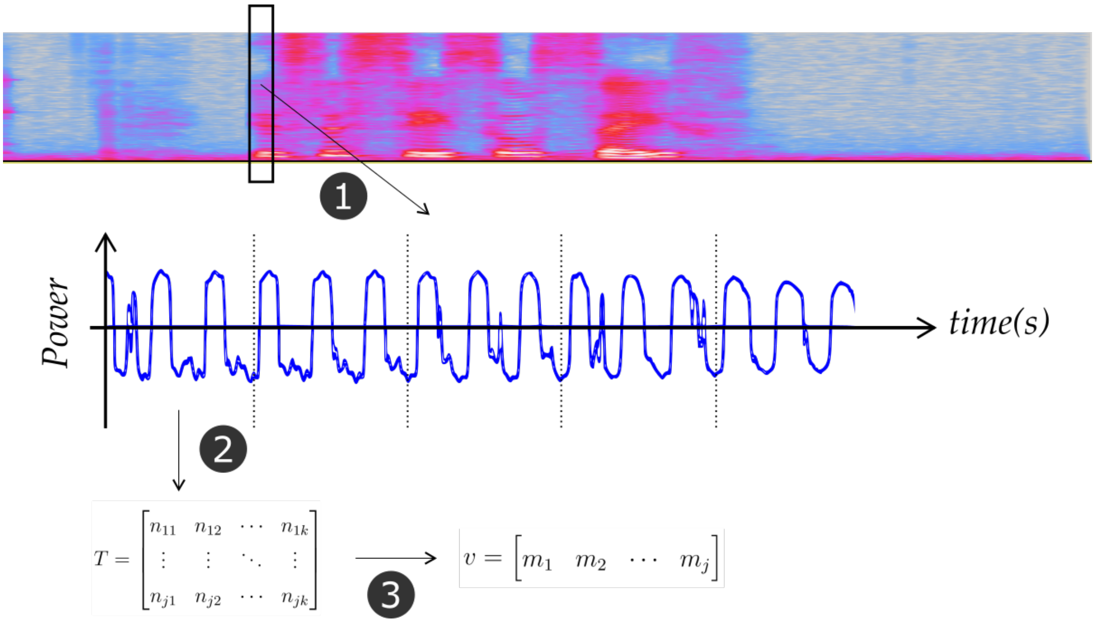

1
Power
time(s)
2
$T = \begin{bmatrix} n_{11} & n_{12} & \cdots & n_{1k} \\ \vdots & \vdots & \ddots & \vdots \\ n_{j1} & n_{j2} & \cdots & n_{jk} \end{bmatrix}$
3
$v = \begin{bmatrix} m_1 & m_2 & \cdots & m_j \end{bmatrix}$

Figure 3: Main steps on the extraction of the phoneme DH from the utterance `fsew0_001`. The spectrum is respective to the prompt “Those thieves stole thirty jewels.”

starting from pre-emphasis to prevent numerical issues and balance the wave up to the Fourier transform and the filter banks finishing with the MFCC. The next section will explain the next step on the data preparation, the clustering and the motivations that led to the choice of method used.

2.3 Data clustering

This section will discuss the clustering method to find expected values for each feature vector v_i and other operations performed to prepare the data are also explained in this section. The problem on classifying the phonemes is that the output of the network is a number, while the phoneme is a character sequence. Therefore, after extracting the network inputs it is necessary to separate them on different subsets. The idea is to separate them on each of the 39 phoneme available.

The process called **Vector Quantization** by (RABINER; JUANG, 1993) is described in four steps. This is a classical technique used in signal processing and the steps for this procedure are:

1. A large set of spectral analysis as vectors $v_1, \ldots, v_n$ with $n \in \mathbb{N}$;
2. A measure of similarity between a pair (v_i, v_k), for i and $k \in \mathbb{N}$;
3. A centroid computation procedure to separate the spectral analysis into M clusters;

4. And a classification procedure for an arbitrary spectral analysis.

The first out of the four steps is done by extracting the features vectors, called by spectral analysis vectors in the Vector Quantization procedure. The following steps can be done by a clustering algorithm based on centroid computation. A clustering algorithm is an process of grouping similar data into sets (ESTIVILL-CASTRO, 2002). There are many algorithms of this type in the literature. However, in this work we will be restricted to the speech recognition literature. In (HASSANZADEH; FAEZ; SEYFI, 2012) and (EL-WAKDY et al., 2008) the authors used a subtractive clustering which is a method based on the mountain clustering. Besides, the work (DAHL et al., 2012) clustered the features using three different criteria, the Maximum-Likelihood (ML), Maximum Mutual Information (MMI) and Minimum Phone Error (MPE). On the other hand, the author of (JAITLY et al., 2012) use a decision tree based clustering method. Table 2 shows the works and the clustering methods are summarised.

Paper	**Clustering method**
(HASSANZADEH; FAEZ; SEYFI, 2012)	Subtractive Clustering
(EL-WAKDY et al., 2008)	Subtractive Clustering
(DAHL et al., 2012)	ML, MMI and MPE
(JAITLY et al., 2012)	Decision Tree Clustering

Table 2: Clustering algorithms used in the Speech Recognition Field.

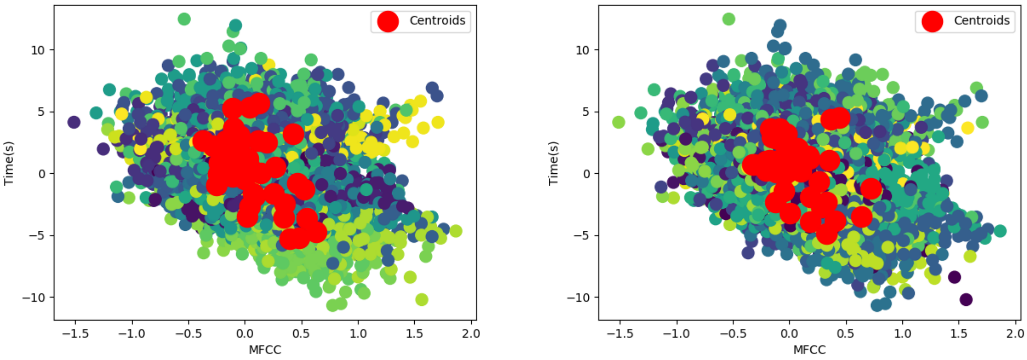
For this work, the clustering algorithm used will be the **K-Means**. This is a centroid based algorithm, also known as Lloyd's algorithm and its first use date to 1967 by (MACQUEEN et al., 1967). The complexity of this algorithm is NP-Hard for the classical Euclidean based similarity distance, as defined in (GAREY; JOHNSON; WITSENHAUSEN, 1982). The process of this algorithm consists on three phases, described as:

Initialisation is the process of creating the initial means, or centroids.

Assignment in this step each observation is assigned to the closest centroid with respect to the similarity function. The most well known and classical similarity function is the squared distance, e.g. Euclidean Distance.

Update is the phase where the new centroids are computed.

Then, the **Assignment** and **Update** phases are repeated until a maximum number of steps or a threshold value is reached. Figure 4 shows the results of clustering the extracted features used in this work. After finding the centroids with K-Means, then we need to



(a)

(b)

Figure 4: Clustering results using 39-Means for instances extracted with 4 cepstrums 4a and 7 cepstrums 4b.

convert the input to an one dimensional value. This is necessary to be able to compare it with the single output value of the ANFIS. Therefore, given a centroid $C = (c_1, \ldots, c_n)$ where n is the dimension of the inputs, then the desired output is the projection of C on the first axis as the desired output for the network, that is c_1.

This chapter presented the main databases for the problem of speech recognition as well as the main feature extraction techniques used and introduced the concepts of clustering data, the main clustering algorithms used in the speech recognition field and discussed the method used in this work. As well as the results of the vector quantisation steps for the data. The next section will present the structure and the basic concepts behind the neural network that will be used in this work, in addition to briefly discuss some classifiers which are commonly used for speech recognition.

3 Into the classification of speech data

The advancements in the field of artificial intelligence is providing an increasing number of works about ASR. However, only recent researches on NN and Deep Neural Networks (DNN) started being applied to ASR. These investigations are resulting in better outcomes compared to previous works, where the GMM-HMMs predominated for a long time. These recent researches are giving good outcomes compared to the previous literature. This chapter will briefly describe a literature review on the ASR field in the period of 2012 and 2017 and explain introductory concepts for understanding the creation and execution of the network develop in this work.

Most of the work in this period are an investigation of DNN or NN for ASR. These networks used around three or four hidden layers, with approximately 1500 and 2000 neurons in each hidden layer. The studies, for this period, have heterogeneous input and output layers, varying in number of neurons and activation functions. For instance, in (SELTZER; YU; WANG, 2013) the authors used approximately 1500 neurons on the output layer, while (YU et al., 2013) used 5976. Most of them also differ in the methods of initialisation of the network, where some are supervised or unsupervised. Table 3 show a resume of each work and their respective best classifier, dataset and outcome.

Table 3: Resume of the works from the last 5 years.

Paper	**Classifier**	**database**	**Error**
(DAHL et al., 2012)	DNN with 3 hidden layers and 2000 units using Triphone Senones	Training set with 32.057, development set with 8.777, and test set with 12.758 (Bing Mobile Voice Search).	30.4%

(GRAVES; MOHAMED; HINTON, 2013)	Pre trained RNN Transducer with 3 hidden layers and 250 senoes each.	Training set with 462 speaker with SA, 50 speaker development set, and 24 speaker test set (TIMIT)	17.7%
(ABDEL-HAMID; DENG; YU, 2013)	CNN with three hidden layers, where each layer is either a 1000-node hidden layer or a pair of convolution and pooling.	462 speakers with no repeated sentences for the test set and 50 speakers for the development set (TIMIT).	20.5%
(SELTZER; YU; WANG, 2013)	MLP with 2048 senones 5 hidden layer, and 1206 output layer using noise-aware and dropout training.	7137 training set utterances from 83 speakers and 330 utterances from 8 speakers for test set (Aurora 4 and WSJ10).	12.4%
(YU et al., 2013)	CD-DNN-HMM with three 2048-neuron hidden layer and 1509-neuron output layer	XBOX Live Search	20.7%[1]
(JAITLY et al., 2012)	DBN pretrained Hybrid 2560-node hidden layer ANN/HMM	Voice Search with 5780 hours of speech.	16.0%
(ABDEL-HAMID et al., 2014)	Convolutional Neural Network with 3 hidden layers and 1000 senones each.	462 speakers training set without identical sentences (TIMIT)	20.07%
(ROWNICKA; RENALS; BELL, 2017)	Very Deep Convolutional Neural Network with 13 hidden layers (Convolutional) and	15 hours of clean speech with 7138 utterances from 83 speakers	7.75%

Besides, in (SELTZER; YU; WANG, 2013), the DNN is investigated for ASR and noise robustness, where the author compares the outcomes of GMM-HMM and the DNN combined with three different approaches to feature enhancement, in particular the

Dropout introduced by (HINTON et al., 2012b). This research obtained a word error rate (WER) of 13%, about 10% bellow than GMM-HMM. Another application of DNN for ASR is shown by (DAHL et al., 2012). This author uses a Context Dependent DNN for Large Vocabulary Speech Reconigtion (LVSR) to improve the performance of ASR system for real world scenarios, where the common system lag behind the human level performance. The authors used Deep Belief Network (DBN), being it a Restricted Boltzmann Machine (RBM) or Gaussian Restricted Boltzmann Machine (GRBM), for pre-traning the proposed hybrid CD-DNN-HMM model. They also tested the DNN with one to five hidden layers, achieving a the maximum accuracy of 70.3% with five. Note that the first (SELTZER; YU; WANG, 2013) and the last work (DAHL et al., 2012) have different purposes and goals, where (DAHL et al., 2012) was executed in real-world scenarios with multiple speakers. Moreover, the paper (YU et al., 2013) introduces a new adaptation technique CD-DNN-HMM models called Kullback-Leibler Divergence (KLD) Regularization. The work made by (YU et al., 2013) achieved an improvement on recognition accuracy with KLD Regularization of approximately 30% relative WER.

Furthermore, in (GRAVES; MOHAMED; HINTON, 2013) a Deep Recurrent Neural Network (DRNN) is investigated for ASR systems. The authors obtained a Phone Error Rate (PER) of 17.7% with a three hidden layer with 250 neurons DRNN pre-trained with Transducer and after 144 epochs. However, no fuzzy related work was found on the period of 2012 and 2017 for the speech recognition field. The most recent work (ZENG; LIU, 2006) and (AVCI; AKPOLAT, 2006) are dated from 2006. In (AVCI; AKPOLAT, 2006) a ANFIS is used to achieve 8% WER. Both works related to fuzzy systems have used a very restricted database, although AVCI has a decent number of utterances only 100 words were available for each speaker resulting in 2000 samples.

This section has described the state of the art for the Speech Recognition field. The most well known works in this period have been detailed in this section, specifying the classifier used, the best accuracy and their main goals. Besides, the outcomes of another research for ANFIS related works in the field of Speech Recognition is explained. The following section will introduce some important concepts on Fuzzy Logic to better understand the system developed in this work.

3.1 Fuzzy logic for speech recognition

Fuzzy Logic is an attempt to better represent the imprecision and vagueness of natural language, as described in (NGUYEN; WALKER, 2005) this is the primitive concept of fuzzy. The idea is to better mimic the human thinking with a many-valued logic. The use of fuzzy logic has been increasing with the technological advancements since they can better represent the meaning of the data as a range of values like a color selector, rather than just a simple answer "Yes" or "No". The next section will introduce the concept of membership functions and how to create fuzzy sets.

3.1.1 Fuzzy Logic Components

The most primitive unit in fuzzy logic is the **linguistic variable**. This is a concept defined in the universe of discourse that can be represented in a range of values (COPPIN, 2004) like *temperature* can be specified as cold, warm and hot.

Moreover, the purpose of this logic is to reason about the **fuzzy sets**. Taking into consideration the fuzzy logic idea of range representation, then a fuzzy set element pertinence follow the same concept. A fuzzy set is nothing more than a function f where its codomain is in the range $[0, 1]$ rather than restricted to $\{0, 1\}$ (NGUYEN; WALKER, 2005).

Definition 3.1.1. Fuzzy Set A fuzzy set is of a set U is a function $U \to [1, 0]$.

In fuzzy logic the linguistic variables are transformed into **membership functions**. These functions will compute a membership degree of a given value. Therefore, the membership function will evaluate the 'compatibility' of an element. Furthermore, as pointed in (NGUYEN; WALKER, 2005), the membership function calculate the "compatibility" of an value rather than the "truth". Then, we say that $A : U \to [0, 1]$ is the membership function of the linguistic variable A, hence $A(u)$, or $\mu_A(u)$, is the **membership degree** of $u \in U$ for the fuzzy set. Then, by representing this sentence as a set of membership functions we can now demonstrate this sentence almost without loss of data.

This section presented the basic concepts of fuzzy logic, defining and comparing them with the classical bivalent logic. The next section will introduce the inference concepts of fuzzy logic, as well as the inference systems.

3.1.2 Fuzzy Inference Systems

As said before, this section will introduce the fuzzy rules, how the fuzzy implication is done and finish with the fuzzy inference systems.

The bivalent logical rules can be expressed as **IF a THEN b** because the logic value of a and b is either True or False. On the other hand, fuzzy rules has to be expressed in the format

$$\text{IF a} \triangle \text{v THEN b} = \text{u}$$

where $\triangle$ is and logic operator ($<$, $=$ or $>$), v and u are crisp or linguistic values. Therefore we can build a set of fuzzy rules and use their outputs to produce an action. This is the most common implication model in the fuzzy literature, the Mamdani (NGUYEN; WALKER, 2005) implication model, which is in contrast with the classical Gödel (MAGNUS, 2014) implication. Moreover, other inference systems like the Tsukamoto FIS and Takagi & Sugeno FIS are used. The Mamdani model allows to receive a set of crisp values an apply fuzzy operations on them to result in a single crisp output as a recommendation or action.

Furthermore, we can define a Fuzzy Inference System (FIS) as an expert system, thus “a process that takes a set of values and uses fuzzy operations to output a value or recommendation” (COPPIN, 2004). Figure 5 shows the general structure of an FIS. The process to defuzzify the output is the most important and simple step in this context. For that, it is necessary to compute **center of gravity** of the intersection of the areas below the graphs at the membership degree of each membership function. Besides, the inference engine has to combine the values in some way, for that an fuzzy operation is used.

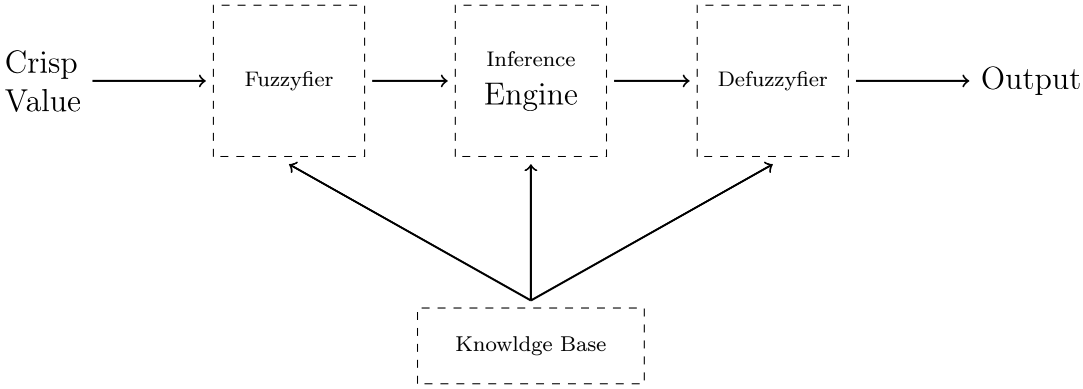


Figure 5: Generic structure of a Fuzzy Inference System.

Therefore, the process of implication on fuzzy set can be summarised, as indicated in (COPPIN, 2004), like:

1. Relate the inputs to the fuzzy sets;
2. Evaluate each case for every fuzzy rule;
3. Combine the information from the rules;
4. Defuzzify the results.

The combination step is usually done by applying **t-norm** or **t-conorm** operations to the fuzzy values. A t-norm operation is defined by (NGUYEN; WALKER, 2005) as a model to the "and" connective, while a t-conorm operation is a model to the "or" connective. Besides, an operation $\Box$ is t-norm if it satisfies the *commutativity*, *associativity*, has 1 as *identity element* and *monotonicity* properties.

Definition 3.1.2. A binary operation $\Box$ is an t-norm if it satisfies

a) $1 \Box x = x$ (1 as identity)

b) $x \Box y = y \Box x$ (Commutativity)

c) $x \Box (y \Box z) = (x \Box y) \Box z$ (Associativity)

d) $\textit{if } a \leq x \textit{ and } b \leq y \textit{ then } a \Box b \leq x \Box y$ (Monotonicity)

Nevertheless, a t-conorm operator is the dual of an "and" operation, then it follows from the De Morgan's Law that an operator $\Diamond$ is a t-conorm if and only if it satisfies *commutativity*, *associativity*, has 0 as *identity element* and *monotonicity* properties. Therefore, the operation $\Diamond$ has to satisfy properties b, c and d from Definition 3.1.2 and e) $0 \Box x = x$ (0 as identity).

Although a very interesting expert system, the FIS has some drawbacks. One disadvatange is the necessity of an external expert to feed the rules for the system, which may not always be available. Furthermore, after the knowledge is fullfiled the system becomes static and can not learn with the data thus representing the expectations of the expert at the moment that the system was built (COPPIN, 2004).

This section introduced the shape of fuzzy rules and the alternative fuzzy inference system designed by Mamdani (MAMDANI; ASSILIAN, 1975). Besides, the structure of a FIS was explained as well as the steps to the inference. Moreover the concepts behind t-norm operators were discussed in this section. The next section will discuss the combination of FIS and the Neural Networks and how they can lead to better results rather than separated.

3.2 Fuzzy Neural Networks

As previously said, this section will introduce the concept of Fuzzy Inference Systems that learn and how that combination leads to interesting results. A Regular Fuzzy Neural Networks (FNN) is a Neural Network with fuzzy signals and/or weights (BUCKLEY; HAYASHI, 1994). The authors of (BUCKLEY; HAYASHI, 1994) and (JANG, 1993) define an FNN in three ways. The first is FNN$_1$, which has real numbers as inputs and fuzzy weights. The second is the FNN$_2$ that has fuzzy set inputs and real number weight. Finally, the third is the FNN$_3$ that uses fuzzy sets as inputs and weights.

The goal of combining both Neural Networks and Fuzzy Inference Systems is to add the learning capability to the FIS. Therefore, in the general FIS model shown in Figure 5 the **Knowledge base** would be unnecessary because the system itself would build the rules. As a result, the Fuzzy Inference system dos not need an expert to create the knowledge base, while the outcomes of the Neural Network are no longer a black box allowing us to know which rules resulted in the output.

Besides, a Hybrid Fuzzy Neural Network (BUCKLEY; HAYASHI, 1994) can be achieved by applying other operations in the activation functions, like T-Norm (for instance, conjunction or min) or T-Conorm.

Table 4: Fuzzy Neural networks and their Equivalent FIS.

Type	**Input/Weights**	**Equivalent FIS**
FNN$_1$	Crisp/Fuzzy	Tsukamoto Model
FNN$_2$	Fuzzy/Crisp	Mandani Model
FNN$_3$	Fuzzy/Fuzzy	Takagi & Sugeno Model

This section presented the basic concepts around Fuzzy Logic necessary to understand the following sections, classified the three types of fuzzy inference systems and how to achieve the equivalent neural network. The next section will detail the network structure used in this work and how the signals are propagated forward.

3.3 ANFIS: Adaptive Network Fuzzy Inference System

This section will describe the ANFIS architecture used in this work, how the signals are propagated forward and a brief description of the existing learning procedures.

3.3.1 Architecture

The ANFIS structure was created by (JANG, 1993) in the 1990's. They are fuzzy system that can learn and, as shown by the author, they are equivalent to some fuzzy inference systems. So, given the output from the extractor, that is, a vector $v = \{v_0, ..., v_n\}$ where n is the number of windows (e.g. Hamming Window) created, $v_i \in \mathbb{R}$ for $1 \leq i \leq n$, and using fuzzy weights, thus we have an FNN$_1$ as previously defined.

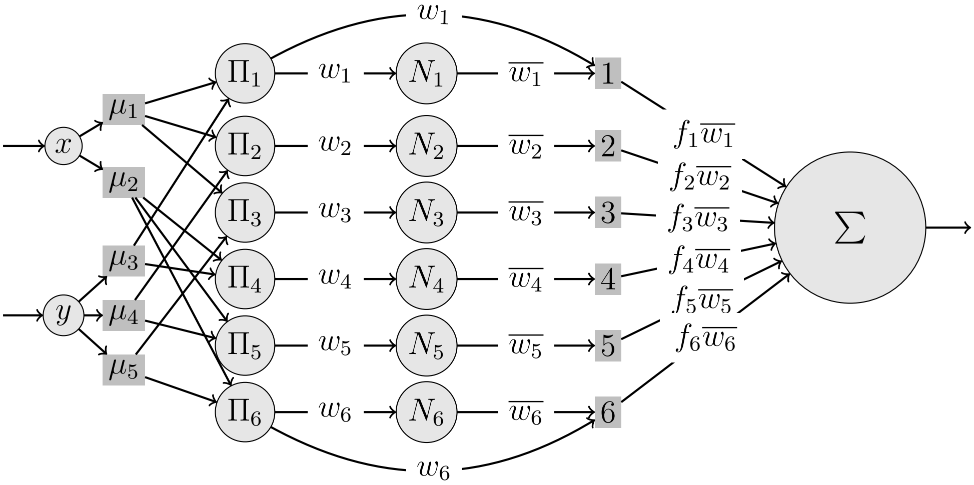


Figure 6: Example of a simplified ANFIS architecture used in this work.

The structure in Figure 6 is a Hybrid System, where a Fuzzy Inference System (Layers one to four) is combined with a Neural Network. More precisely, this structure represent a Type-1 ANFIS which is equivalent to a FNN$_1$ and thus to an Tsukamoto Fuzzy Inference System. Notice that each node in the third layer will receive all the outputs from the second layer, but for a better visualization only one connection is shown in Figure 6.

The ANFIS structure is composed of two types of nodes: adaptive and non-adaptive nodes. The former is represented in Figure 6 by the square nodes. They hold the parameters of the network, which will be updated in the learning procedure. The latter is represented by circle nodes, in the same figure. Although they are represented as nodes in the network, they do not have an activation function and are just performing operations on their inputs. The operations of each node is explained in more detail below.

This combination will result in the union of the qualities and flaws of both Neural Networks and Fuzzy Inference Systems. However, the weakness of one can be the strength of the other, as said before, where the FIS will now be capable of learning and the NN will no longer be a black box as well as improving its capacity of data representation with the fuzzy capacity to deal. Now, we will specify how each layer works in this structure, following the definitions given in (JANG, 1993). The description of the layers behavior was adapted from (AVCI; AKPOLAT, 2006). For that, consider the structure in Figure 6 and a set of precedent parameters p_i, q_i, and r_i where $i = 1, \ldots, 5$. Besides, given an adaptive network with L layers, the i-th node on j-th layer can be expressed as (i, j), therefore O_j^i is the output of i-th node in the j-th layer.

Layer 1 is the fuzzifier layer. This layer is composed of premise fuzzy sets and will fuzzify the crisp input values. Thus, every node in this layer has a membership function. Usually, a bell-shaped function varying from 1 to 0 is used, but any piecewise differentiable function can be used (JANG, 1993). Therefore, the output of this layer is the result of the membership value for the input, that is

$$O_1^i = \mu_A(x) \tag{3.1}$$

Commonly used membership functions are:

$$bell_1(x) = \frac{1}{1 + \left[\left(\frac{x-c}{a}\right)^2\right]^b} \tag{3.2a}$$

$$bell_2(x) = exp\left(-\left[\frac{x-c}{a}\right]^2\right) \tag{3.2b}$$

Layer 2 is the layer that computes the firing strength of each rule of the system. This stage is done by using a T-norm operator which perform a generalised AND (i.e. product or min). In Figure 7 a product operation is computed for the inputs i and j.

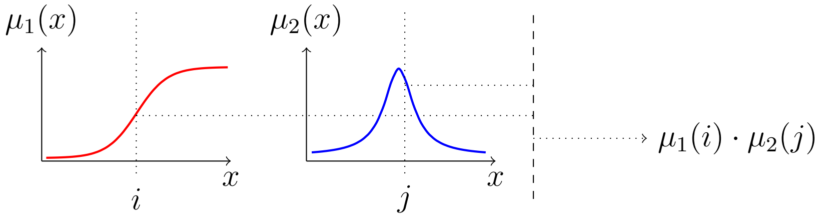


Figure 7: Visual outcome of product operation.

Layer 3 is responsible for calculating the ratio of the fire strength of a rule. This can be calculated as a simple mean value. This layer and the previous have only non-adaptive neurons, as explained in above. For instance, we could compute the the layer output with equation 3.3.

$$O_3^i = \overline{w_i} = \frac{w_i}{w_1 + \ldots + w_6} \tag{3.3}$$

Layer 4 is the consequent layer, it is composed of the consequent fuzzy sets. For instance, the previous output, given in the example on Layer 3, would activate the node for Average linguistic variable in this layer. Each node in this layer is an adaptive node, so they hold a set of *consequent* parameters, for instance a, b and c.

$$O_4^i = \overline{w_i} f_i = \overline{w_i}(a_i x + b_i y + c_i) \tag{3.4}$$

For the FNN_2 and FNN_3, the equation 3.4 is used directly, because the fourth layer inputs are already given as a linear combination of the inputs and consequent parameters. However, for FNN_1 he output of this layer is given by feeding the layer inputs into a monotonic function. This monotonic function is also considerate as a membership function and has some implications on the behavior of the learning procedure explained in Section 4.1.

Layer 5 is the output layer. This layer will receive the values from the consequent layer, and will compute the overall output, or **infer** the result using a centroid, or summation of all incoming values.

$$O_5^1 = \frac{\sum_i w_i f_i}{\sum_i w_i} \tag{3.5}$$

This section presented an introduction to fuzzy concepts necessary to better understand the structure used in this work. Explaining the concept of the smallest unit of work, the membership functions, and continuously expanding them. The definitions and theory of fuzzy sets and fuzzy expert systems is also discussed in this section, finishing with the description of the architecture used in this work. Besides, this chapter has shown a brief review of the state of the art on speech recognition, detailing their classifiers; databases and accuracy. At the end an description of the data flow in an simplified version of the network is given. In the next chapters the main steps of creating the learning procedure for the proposed network are detailed, finishing the theory used in this work.

4 Using Fuzzy and Neural Networks for Speech Recognition

The Fuzzy Networks, thus the adaptive fuzzy inference system, can learn by using classic back-propagation and gradient descent learning algorithm, although (JANG, 1993) assert that this can be slow. The author suggested some alternatives to the classic back-propagation algorithm by implementing online and offline algorithms which make use of Least-Square Estimation (LSE) to compute the consequent parameters. In this chapter the *hybrid online leaning procedure* is specified by detailing the requirements that the model needs to implement this procedure and how the parameters in the network are updated.

4.1 Pattern Learning: Hybrid Online Learning Rule

This method is used by most most of the works with an ANFIS, for instance (AVCI; AKPOLAT, 2006) and (ZENG; LIU, 2006) used this method for speech recognition. In order to implement a learning procedure, the definition of an overall error measure is necessary. For this, the equation 4.1 will be used. Consider $p \in P$, where P is all the input-output pairs given to the network.

$$E_p = \sum_{m=1}^{L} \left(T_{m,p} - O^L_{m,p}\right)^2 \tag{4.1}$$

where L is the layer, $T_{m,p}$ is mth component of the pth output vector, and $O^L_{m,p}$ is mth component of the actual output vector produced by the pth input vector (JANG, 1993). Therefore, to correctly update each parameter we need to compute the gradient for the error function, which will be computed with

$$\frac{\partial E_p}{\partial \alpha} = \sum_{O^* \in S} \frac{\partial E_p}{\partial O^*} \frac{\partial O^*}{\partial \alpha} \tag{4.2}$$

that is the derivative of equation 4.1 by the chain rule. Therefore, the equation 4.3 is learning rate for a generic parameter α is given by $\Delta\alpha = -\eta\frac{\partial E_p}{\partial \alpha}$ where

$$\eta = \frac{m}{\sqrt{\sum_{\alpha}\left(\frac{\partial E_p}{\partial \alpha}\right)^2}} \tag{4.3}$$

Equation 4.3 shows that the variable m directly affects the learning rate of the network. Although it can be set statically, it may lead to a slow convergence. In (JANG, 1993) the author describes some heuristic increases for the value of m. Thus, based on his work we propose the following heuristic increase of m is used:

1. If there are four consecutive error **decreases** then m is increased by 10%;
2. If there are four consecutive error **increases** then m is decreased by 10%.

The updating procedure, or learning procedure is based on being able to separate the precedent parameters from the consequent. If we take a look at the overall output of the network in Equation 3.5, the fifth layer, is the sum of the outputs in the fourth layer. If we can write the output of the fourth layer (fi) as a linear combination of the consequent parameters, then for each p we can write.

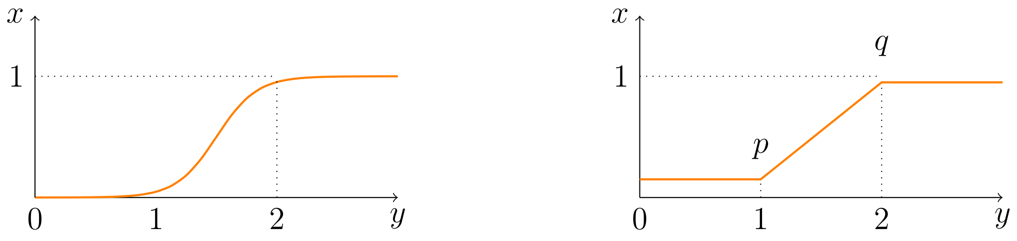
$$\begin{aligned}
\sum_{i}^{k} \overline{w_i} f_i &= \frac{\sum_i w_i f_i}{\sum_i w_i} \\
&= \frac{w_1 f_1 + w_2 f_2 + \cdots + w_k f_k}{w_1 + \cdots + w_k} \\
&= \frac{w_1 f_1}{w_1 + \cdots + w_k} + \cdots + \frac{w_k f_k}{w_1 + \cdots + w_k} \\
&= \overline{w_1} f_1 + \cdots + \overline{w_k} f_k \\
&= \overline{w_i}(a_1 x + b_1 y + c_1) + \cdots + \overline{w_k}(a_k x + b_k y + c_k) \qquad (4.4)
\end{aligned}$$

where k is the number of rules of the network. So, if we rewrite the last equation by putting the consequent parameters in evidence and switching the left side by the expected value of p, we have

$$\begin{aligned}
T_p &= \sum_{i}^{k} (\overline{w_i} x) a_i + (\overline{w_i} y) b_i + (\overline{w_i}) c_i \\
T_p &= (\overline{w_1} x) a_1 + (\overline{w_1} y) b_1 + (\overline{w_1}) c_1 + \cdots + (\overline{w_k} x) a_k + (\overline{w_k} y) b_k + (\overline{w_1}) c_k \qquad (4.5)
\end{aligned}$$

Note that up to the layer three all the values in equation 4.5 are known, except the consequent parameters. Therefore, it is possible to build a linear system and find an approximation for each consequent parameter. This suggests that using a technique to solve this system can be used on the learning procedure.

Although the Equation 4.5 can be easily achieved on FNN_3 and FNN_2 it is not true for the type used in this work. The Type-1 ANFIS has a monotonic function in the fourth layer, as said in the previous section, then to create this linear system for Type-1 it is necessary to replace the monotonic function for an piecewise linear approximation (PLA). Figure 8 shows an example of how to simplify this function. Although there are techniques acquiring a PLA version of a function f, they are not necessary to accomplish the requirements of the online learning. Since the values of the consequent parameters p and q, as shown in Figure 6, will be given by the learning procedure, and we can set maximum and minimum values of x as close as possible to 1 and 0, then we can easily write the equation f' as the PLA version of f. Furthermore, as this function is discretised we can also apply Equation 4.5 on the subject of this work.

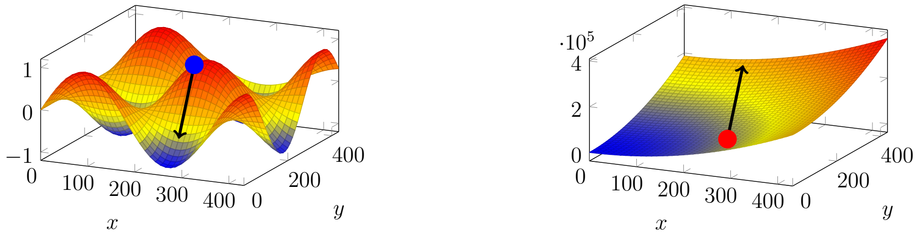


(a) Continuous monotonic function.

(b) PLA of monotonic function.

Figure 8: Example of monotonic function (8a) and its PLA (8b).

Now, we can define the hybrid online learning rule. This learning procedure consists on feeding the signals forward until layer three while the consequent parameter set is fixed. After this point, the precedent parameters are considered to be fixed and the linear system is built to find an approximation for the consequent parameters. When the consequent parameters are updated, the error is propagated backwards to the adaptive nodes by the Gradient Descent Method, while the consequent parameters are considered to be fixed because they are the best value possible for the given inputs since they were computed with LSE.



(a)

(b)

Figure 9: Graphical representation of the classic gradient descent method.

The Gradient Descent Method consists on computing the impact of each node output on the error of a given input, considering the next layers and a error function and updating parameters (or weights) by a a factor computed by the partial derivative of the error function with respect to each node output. This method is represented by the solution space in Figure 9, where the initial point is marked as the blue dot while vector coming out of it represent the direction of a possible best solution, as a minimum value for 9a and maximum value for 9b.

Therefore, the proposed model consists on the feature extraction using MFCC and a Type-1 ANFIS as the classifier with a midway unsupervised learning method. The Figure 10 show an visual example of the proposed model, where w_p is the wave of the phoneme p for a speaker, v_i is the feature vector obtained from the extraction method, o_i is the expected output computed by the K-Means and $p^{'}$ is the phoeneme classified by the Type-1 ANFIS. At the extraction block, each node is a extraction stage as explained in Section 2.2. Besides, in the classification block, the Type-1 ANFIS represents each step of the learning procedure to classify a phoneme p.

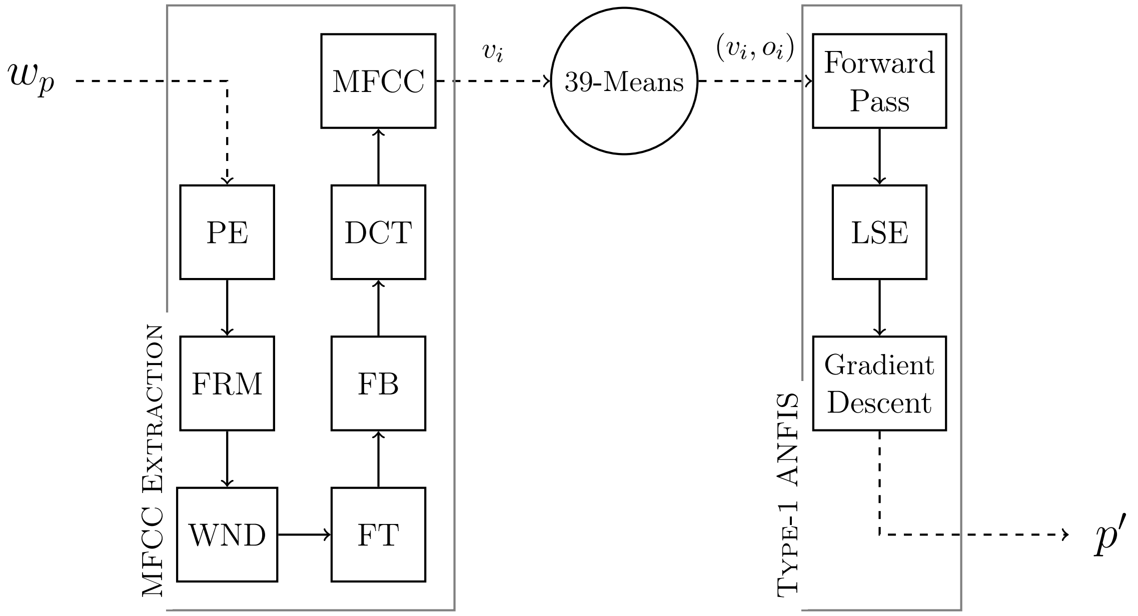


Figure 10: The acoustic model proposed in this work.

In this section we presented how to calculate the updating value for each precedent parameter of the network, also we detailed the heuristic increase of the m factor. Besides, we described the step of separating the parameter space in two disjoint sets, which is important to allow us the application of LSE. Moreover, we presented how to generate an piecewise linear approximation of the consequent function for Type-1 ANFIS and thus creating a consequent parameter set for this network.

4.2 Least Square Estimation

This section will detail how we applied an iterative version of the Least Square Estimation technique to be used on the learning procedure. Besides, a brief definition of the mathematical problem of modeling unknown systems is described. Equation 4.5 presents a step in the unknown system problem, called **Structure Identification**. In (JANG; SUN; MIZUTANI, 1997), the authors separate this problem in two phases:

1. **Structure Identification**: consists on finding a class of suitable models to be conducted. Usually, it is done by denoting a parameterised function $f(x;\gamma)$ where γ is the parameter vector. Considering the example given in Section 3.3.1, then $\gamma = \{a, b, c\}$;

2. **Parameter Identification**: the structure is known, and an optimisation technique is applied to find the vector $\hat{\gamma}$ where $f(x;\hat{\gamma})$ can describe the model.

The idea of using the Least Square Estimation method is to identify the parameters that best fit the model, that is structured target system. The LSE is a regression method that has been widely used to solve over determined systems. In (MISHRA, 2005), the authors used the method to find Harmonic Estimations, while (VAHIDI; STEFANOPOULOU; PENG, 2005) used this technique to estimate vehicle mass and grade. As shown in (JANG, 1993), the convergence happens faster when using LSE rather than other conventional methods presenting both online and offline versions of the algorithm. Since the first phase of the model determination was shown in Section 4.1 with the equation 4.5, we will focus on the optimisation technique used in this work. After structuring the model like Equation

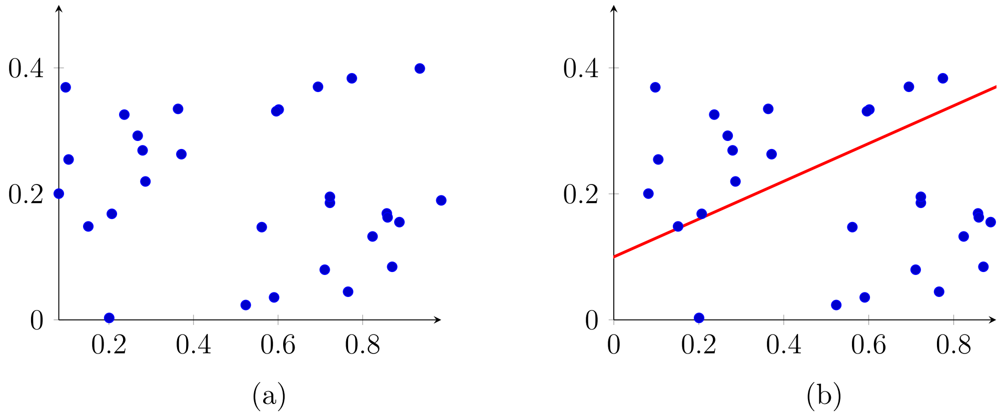


Figure 11: Example of a linear regression done by LSE.

4.5, that is, separating the parameter set in two disjoint sets of consequent and precedents parameters, we can write it as the matrix equation $AX = B$. More precisely

$$A_{P\times M}X_{M\times 1} = B_{P\times 1} \tag{4.6}$$

where M is the number of consequent parameters of the system.

Equation 4.6 is an over determined problem, since P is the number of inputs/outputs from the training set and is generally bigger than M. Therefore, we can determine an *least square estimation* X^* for which the squared error $||AX - B||^2$ is minimised. For this, we can write $X^* = (A^TA)^{-1}A^TB$ where $(A^TA)^{-1}$ is the pseudo inverse of A. However, it is computationally expensive to calculate the inverse and it may lead to problems if A^TA is singular. Therefore, an iterative approach is better. In (JANG; SUN; MIZUTANI, 1997), the authors referenced an well known iterative method based on Equation 4.7.

$$\begin{cases} X_{i+1} &= X_i + S_{i+1}a_{i+1}(b_{i+1}^T - a_{i+1}^T X_i) \\ S_{i+1} &= \frac{1}{\lambda}\left[S_i - \frac{S_i a_{i+1} a_{i+1}^T S_i}{\lambda + a_{i+1}^T S_i a_{i+1}}\right] \end{cases} \tag{4.7}$$

where S_i is *covariance matrix* and $S_0 = I\gamma$. The value of γ will proportionally increase the precision of the LSE but values too high will make it amiss the classification, a_i^T is the row vector of A, b_i^T is the ith element of B. Besides, λ is a value between 0 and 1 called *forgetting factor* that will make the older parameters to slowly fade as the new inputs/outputs are fed to the network.

This chapter introduced the basic concepts that leads to the creation of the learning procedure. Detailed how to convert the fourth layer of the FNN_1 so that it can also implement the hybrid learning rule. Defined how the least square estimation can be used to update the consequent parameters and helping the network to a faster convergence and characterised the steps to the mathematical modeling of unknown systems. The next section will present the outcomes of the work and briefly discuss the goals for future work.

5 Discussion of results and analysis

This chapter will present the results of this work on the MOCHA-TIMIT dataset, compare them with the state of art and briefly discuss the outcomes. The pre-processing consisted in applying the MFCC feature extraction in each portion that represented a phoneme by considering the label file provided. The number of Cepstrums used where 4 for each speaker and each feature represents the mean of the values in the respective Cepstrum. The small number of features used is to prevent memory issues, since the number of rules in the network is given by the product of the size of each fuzzy set $|\mu|$ and input i, that is $\prod_i^n |\mu|$.

The dataset consists on two folders, one for each speaker. The folder `fsew0` is for the female speaker, while `msak0` for the male. Thus, in Table 5 `fsew0H` stands for the results on this data with the heuristic increase value of m. For all the tests, we have used a forgetting factor λ of 0.9 and a initial γ of 1000 for the Least Square Estimation. Besides we have used the membership function described by Equation 3.2b for every test since the membership function described by Equation 3.2a were resulting in numerical issues. The architecture variation was on the number of cepstrums (inputs) and the number of membership functions per input, as shown in Table 5 by MF2 and MF3. For convergence, the maximum number of epochs used was 250, while the tolerance was 10^{-5}.

The Table 5 shows the results for the number of cepstrums, as explained in previous sections, the size of the feature vector. The MF2 and MF3 stands for the number of membership functions for each input, therefore the Mean Squared Error (MSE) on 4 cepstrums for MF3 is the total error for an ANFIS with 12 precedents and 81 rules, that is 12 nodes on input layer and 81 nodes on the hidden layers.

Therefore, based on the results shown in Table 5, the Phone Error Rate on all the tests where too high. Even varying the parameters on the network, like the number of rules that raised from 16 to 81, respectively from MF2 to MF3, the PER is still high and even increased in some cases. However, the Mean Squared Error of the network is really low and thus most of the outputs on the network did got close to the expected value. Although, as shown in Section 2.3 the data is high correlated even after the Discrete Cosine Transform from the feature extraction.

		MSE		$\overline{time}$		$\overline{epochs}$		PER	
Test data	Cepstrums	MF2	MF3	MF2	MF3	MF2	MF3	MF2	MF3
		%	%	s	s			%	%
`fsew0H`	4	3	3	0.129	0.92	193	194	94	94.2
`fsew0`	4	4	4.5	0.2	0.98	230	239	93.7	94.2
`msak0H`	4	2.5	X	0.17	X	194	X	94.6	X
`msak0`	4	X	X	X	X	X	X	X	X

Table 5: Resume of results on each test set. Test sets appended with `H` were executed with heuristic increase of m.

With respect to the heuristic variation of m, it is possible to see that the network did converged in less epochs than when the static value of m was used. Consequently, the time for each sample from `fsew0H` to `fsew0` for the MF3 did decreased. However, no significant difference is seen from the MSE when compared with the use of the heuristic increase, as well as the PER. Therefore, the use of a heuristic variation of the learning rate only lead to an faster convergence.

Although the results are not as expected there was no changes on either the LSE parameters and the functions of the architecture. Even though the variation on the LSE parameters (λ and γ) may have a small impact on the classification accuracy, since only the forgetting factor could have a change it may not vary too much because it may lead to numerical problems. Nevertheless, a change on the membership function could have a bigger impact given that the function could model better the given system (JANG, 1993). Besides, a different T-norm operator could be used to combine the fuzzy values.

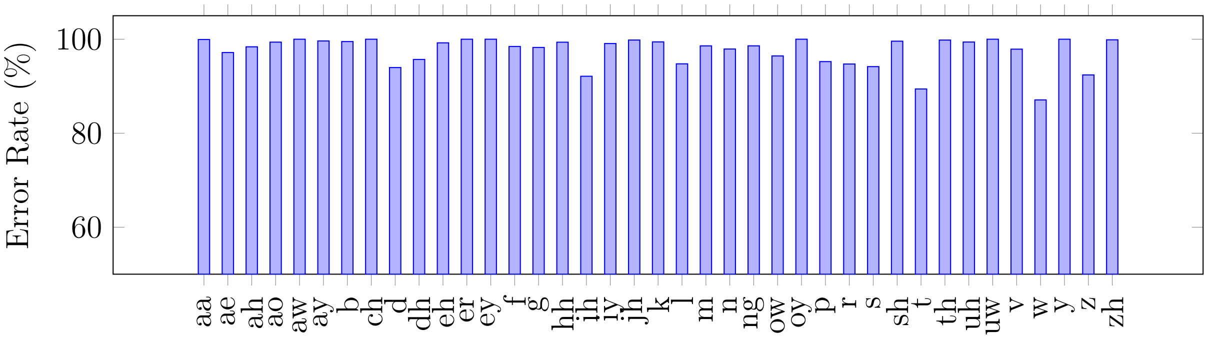


Figure 12: Error rate for every phoneme extracted with 4 cepstrums.

The chart in Figure 12 shows the error rate for each phoneme for all test sets. The results from the bar chart point that phonemes with no closely related sound, like `t` or `w` had a slightly better accuracy, while others like `aa` and `aw` had almost no accuracy. This might be related to the extraction method used, since MFCC is the frequency along the time axis then with a similar sound the frequency also becomes similar. Furthermore, the unsupervised learning procedure used to create a numerical expected value may also have influenced since it was necessary to use a projection of the centroids the values have lost some of the distance perspective, thus making them more closely related.

5.1 Learning rate variation

The learning rate m were configured to initialise with a random value in $[1, 10]$. As explained in Section 4.1, this value is updated by an heuristic strategy. The graphs on Figure 13 where collect while running tests for the phoneme `IY`. Although some sections of the plots seems to be constant, they where varying on less significant decimal places.

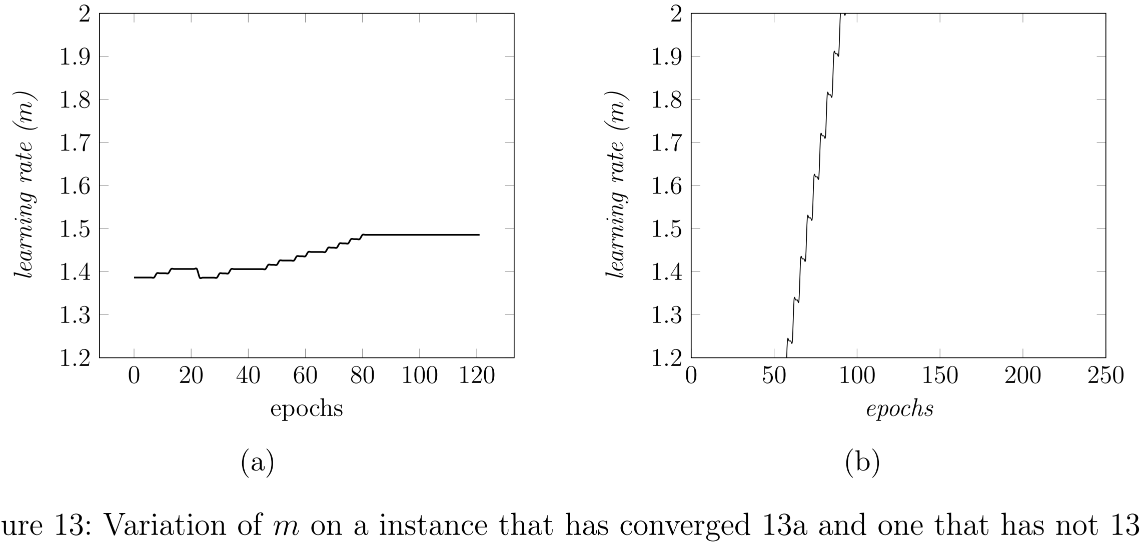


Figure 13: Variation of m on a instance that has converged 13a and one that has not 13b.

Analysing both figures 13a and 13b, it is possible to see that the variation lead to a faster convergence on Figure 13a as the value converged on 120 epochs. Although, in Figure 13b the learning rate variation is constant increasing on 10% for every 4 epochs and the error rate is also constantly decreasing. Therefore, it can also lead to local minimal in some cases, resulting on the wrong phoneme.

This chapter explained the results of this work comparing both heuristic and non heuristic increase of the learning rate, presented the PER for each test set and the MSE. Besides, we also made a briefly discussion about the actions that led to the classification results for all and every phoneme. The next chapter will detail the conclusions of this work and the goals for future research.

6 Final Remarks

This work has presented a proposal for developing a Type-1 Adaptive Network Fuzzy Inference System, based on Tsukamoto Inference System, specifically to perform Speech Recognition. In this context, the work consisted on all the classical phases for Signal Processing, going through the steps of signal emphasis, signal processing with MFCC technique, the data decorrelation with DCT. Then the vector quantisation was explained as well as the contents of the feature vector used in the work, finishing the pre-processing with the clustering techniques used in the ASR field, as well as the one used in this work. Then, after the data pre-processing phase, it was necessary to classify the data into the correct phonemes. For that, it was introduced the concepts of Fuzzy Logic that are necessary to understand the ANFIS behaviour. Then, the structure of an ANFIS is explained and how this system can learn, finishing with the results.

Thus, comparing the results of this work with the state of the art it is possible to conclude that the structure created is not feasible to ASR. Even comparing with the results of an ANFIS for ASR as given by (AVCI; AKPOLAT, 2006) the results are yet too bad. Although, the size of the database used by (AVCI; AKPOLAT, 2006), and the number of tests, was really small compared to the number of samples used in this work, which may have lead the network to over-fitting. For instance, (EL-WAKDY et al., 2008) used a database with 1 speaker, 3 English words and 30 tests. H owever, the author also used a different feature extraction method than MFCC, called Wavelet Packet, and this can imply that the MFCC may not be a good extraction method to use with ANFIS. As a consequence of the limitation on the number of outputs (1) that the ANFIS can have, it was necessary to use a projection of the centroids as expected values to each phoneme. This made the data lose some dimension and thus the distance between them got closer.

The aims for our future works are to improve and test other methods of ASR. In order to improve the results' accuracy, we can work specifically in the architecture to allow a multiple output network, and therefore a Coactive Neuro-Fuzzy Inference System (CANFIS) (JANG; SUN; MIZUTANI, 1997) or other types of ANFIS may lead to better results. Also, other types of feature extraction can be used, since the Wavelet Packet Transform has better results than the MFCC when combined with the ANFIS.

References

ABDEL-HAMID, O.; DENG, L.; YU, D. Exploring convolutional neural network structures and optimization techniques for speech recognition. In: *Interspeech*. [S.l.: s.n.], 2013. p. 3366–3370.

ABDEL-HAMID, O. et al. Applying convolutional neural networks concepts to hybrid nn-hmm model for speech recognition. In: IEEE. *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. [S.l.], 2012. p. 4277–4280.

ABDEL-HAMID, O. et al. Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on audio, speech, and language processing*, IEEE, v. 22, n. 10, p. 1533–1545, 2014.

ACERO, A. Acoustical and environmental robustness in automatic speech recognition. In: *Proc. of ICASSP*. [S.l.: s.n.], 1990.

AREL, I.; ROSE, D. C.; KARNOWSKI, T. P. Deep machine learning-a new frontier in artificial intelligence research [research frontier]. *IEEE computational intelligence magazine*, IEEE, v. 5, n. 4, p. 13–18, 2010.

AVCI, E.; AKPOLAT, Z. H. Speech recognition using a wavelet packet adaptive network based fuzzy inference system. *Expert Systems with Applications*, Elsevier, v. 31, n. 3, p. 495–503, 2006.

BLACKMAN, R. B.; TUKEY, J. W. The measurement of power spectra from the point of view of communications engineering—part i. *Bell Labs Technical Journal*, Wiley Online Library, v. 37, n. 1, p. 185–282, 1958.

BUCKLEY, J. J.; HAYASHI, Y. Fuzzy neural networks: A survey. *Fuzzy sets and systems*, Elsevier, v. 66, n. 1, p. 1–13, 1994.

COPPIN, B. *Artificial intelligence illuminated*. [S.l.]: Jones & Bartlett Learning, 2004.

DAHL, G. E. et al. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on audio, speech, and language processing*, IEEE, v. 20, n. 1, p. 30–42, 2012.

EL-WAKDY, M. et al. Speech recognition using a wavelet transform to establish fuzzy inference system through subtractive clustering and neural network(anfis). In: WSEAS. *WSEAS International Conference. Proceedings. Mathematics and Computers in Science and Engineering*. [S.l.], 2008.

ESTIVILL-CASTRO, V. Why so many clustering algorithms: a position paper. *ACM SIGKDD explorations newsletter*, ACM, v. 4, n. 1, p. 65–75, 2002.

GAREY, M.; JOHNSON, D.; WITSENHAUSEN, H. The complexity of the generalized lloyd-max problem (corresp.). *IEEE Transactions on Information Theory*, IEEE, v. 28, n. 2, p. 255–256, 1982.

GRAVES, A.; MOHAMED, A.-r.; HINTON, G. Speech recognition with deep recurrent neural networks. In: IEEE. *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on.* [S.l.], 2013. p. 6645–6649.

GROUP, S. S. *The CMU Audio Databases.* 1991. Accessed 26-September-2017. Disponível em: <http://www.speech.cs.cmu.edu/databases/an4/index.html>.

HASAN, M. R. et al. Speaker identification using mel frequency cepstral coefficients. *variations*, v. 1, n. 4, 2004.

HASSANZADEH, T.; FAEZ, K.; SEYFI, G. A speech recognition system based on structure equivalent fuzzy neural network trained by firefly algorithm. In: IEEE. *Biomedical Engineering (ICoBE), 2012 International Conference on.* [S.l.], 2012. p. 63–67.

HINTON, G. et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, IEEE, v. 29, n. 6, p. 82–97, 2012.

HINTON, G. E. et al. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

HIRSCH, H.-G.; PEARCE, D. *Aurora.* 1997. Accessed 26-September-2017. Disponível em: <http://aurora.hsnr.de/index-2.html>.

JAITLY, N. et al. Application of pretrained deep neural networks to large vocabulary speech recognition. In: *Thirteenth Annual Conference of the International Speech Communication Association.* [S.l.: s.n.], 2012.

JANG, J.-S. Anfis: adaptive-network-based fuzzy inference system. *IEEE transactions on systems, man, and cybernetics*, IEEE, v. 23, n. 3, p. 665–685, 1993.

JANG, J.-S. R.; SUN, C.-T.; MIZUTANI, E. Neuro-fuzzy and soft computing; a computational approach to learning and machine intelligence. CUMINCAD, 1997.

JUANG, B.-H.; RABINER, L. R. Automatic speech recognition–a brief history of the technology development. *Georgia Institute of Technology. Atlanta Rutgers University and the University of California. Santa Barbara*, v. 1, p. 67, 2005.

KIM, C.; STERN, R. M. Power-normalized cepstral coefficients (pncc) for robust speech recognition. In: IEEE. *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on.* [S.l.], 2012. p. 4101–4104.

MACMILLIAN, C. D. William d. halsey. *Mission Hills, California, USA*, 1986.

MACQUEEN, J. et al. Some methods for classification and analysis of multivariate observations. In: OAKLAND, CA, USA. *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability.* [S.l.], 1967. v. 1, n. 14, p. 281–297.

MAGNUS, P. forall x: An introduction to formal logic. 2014.

MAMDANI, E. H.; ASSILIAN, S. An experiment in linguistic synthesis with a fuzzy logic controller. *International journal of man-machine studies*, Elsevier, v. 7, n. 1, p. 1–13, 1975.

MANJUNATH, T.; HEGADI, R. S.; RAVIKUMAR, G. A survey on multimedia data mining and its relevance today. *IJCSNS*, v. 10, n. 11, p. 165–170, 2010.

MISHRA, S. A hybrid least square-fuzzy bacterial foraging strategy for harmonic estimation. *IEEE Transactions on Evolutionary Computation*, IEEE, v. 9, n. 1, p. 61–73, 2005.

NGUYEN, H. T.; WALKER, E. A. *A first course in fuzzy logic.* [S.l.]: CRC press, 2005.

RABINER, L. R.; JUANG, B.-H. *Fundamentals of speech recognition.* [S.l.]: PTR Prentice Hall Englewood Cliffs, 1993.

RAN, L.; HELAL, S.; MOORE, S. Drishti: an integrated indoor/outdoor blind navigation system and service. In: IEEE. *Pervasive Computing and Communications, 2004. PerCom 2004. Proceedings of the Second IEEE Annual Conference on.* [S.l.], 2004. p. 23–30.

ROWNICKA, J.; RENALS, S.; BELL, P. Simplifying very deep convolutional neural network architectures for robust speech recognition. In: IEEE. *Automatic Speech Recognition and Understanding Workshop (ASRU), 2017 IEEE.* [S.l.], 2017. p. 236–243.

SELTZER, M. L.; YU, D.; WANG, Y. An investigation of deep neural networks for noise robust speech recognition. In: IEEE. *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on.* [S.l.], 2013. p. 7398–7402.

SRIVASTAVA, J. et al. Web usage mining: Discovery and applications of usage patterns from web data. *Acm Sigkdd Explorations Newsletter*, ACM, v. 1, n. 2, p. 12–23, 2000.

VAHIDI, A.; STEFANOPOULOU, A.; PENG, H. Recursive least squares with forgetting for online estimation of vehicle mass and road grade: theory and experiments. *Vehicle System Dynamics*, Taylor & Francis, v. 43, n. 1, p. 31–55, 2005.

VENTER, H.; ELOFF, J. H. A taxonomy for information security technologies. *Computers & Security*, Elsevier, v. 22, n. 4, p. 299–307, 2003.

YU, D.; DENG, L.; DAHL, G. Roles of pre-training and fine-tuning in context-dependent dbn-hmms for real-world speech recognition. In: *Proc. NIPS Workshop on Deep Learning and Unsupervised Feature Learning.* [S.l.: s.n.], 2010.

YU, D. et al. Kl-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition. In: IEEE. *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on.* [S.l.], 2013. p. 7893–7897.

ZENG, J.; LIU, Z.-Q. Type-2 fuzzy hidden markov models and their application to speech recognition. *IEEE Transactions on Fuzzy Systems*, IEEE, v. 14, n. 3, p. 454–467, 2006.