uc3m | Universidad Carlos III de Madrid

University Degree in Telecommunication Technologies Engineering

2020-2021

Bachelor Thesis

Dimensional and Discrete Emotion Recognition using Multi-task Learning from Acoustic features and Linguistic features extracted from Speech

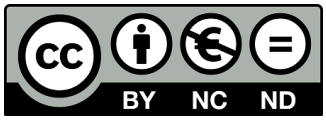
---

Martín Iglesias Goyanes

Tutor: Antonio Artés Rodríguez

Department of Signal Theory and Communications

Madrid, 23 June 2021



ABSTRACT

The majority of research in speech emotion recognition focuses on the classification of discrete emotions either from acoustic features or text features. This thesis demonstrates that the dimensional representation of emotions is also very valuable and it shows its advantages over categorical emotions. The thesis proposes two different systems which both use bimodal features (text and acoustics) in order to recognize discrete and dimensional emotions. A sequential system that first performs dimensional regression and then classification and a parallel system that performs classification and regression at the same time.

The thesis develops a multi-task regression model that serves as the core for both systems. Using the Concordance Correlation Coefficient (CCC) for evaluation it is discovered that the thesis developed architecture for dimensional regression outperforms across all dimensions (valence, arousal, dominance) the regression model introduced in previous research at the Cambridge institution. In addition, the thesis proves that the sequential system outperforms the parallel system in the recognition of both discrete (classification accuracy) and dimensional emotions (CCC). This finding verifies the validity of the theoretical model of psychology about dimensional emotions and its ability to represent a discrete emotion in a three dimensional space without losing any information. Furthermore, it is demonstrated how transfer learning can be used in this specific task to improve the results of the system.

Keywords: Speech emotion recognition, Affective computing, Transfer learning, Deep learning, Audio processing, Text processing, Machine learning

DEDICATION

To Antonio Artés-Roderíguez, for his great expertise, invaluable advice and input, and spur this project forward.

To Angela Moreno Martínez and her colleagues at the Signal Theory and Communications Department at UC3M, for providing continuous help, advice and support when it was required.

To the professors and staff members at Universidad Carlos III de Madrid, the University of Maryland and Miraflores High School in Oleiros for the excellent education I have received and always provide me with insightful answers to my many questions and doubts. Special thanks to Professor James Reggia, for walking me through my initial steps into the world of machine learning and data analysis, which have resulted in this project.

To my friends in A Coruña, Madrid, Leganés, Maryland, and all over the world, for always being there to count on over the past four years of this life journey. Lucía, Carmen, Alicia, Esther, Pablos, David, Iván, Celia, Tomás, Alberto, Teun, Manueles, Juan, David, Jaime, Álvaro, Jerónimo .... You have put up with my unsolicited jokes and particular sense of humor, my exceeding outspoken personality and other flaws of mine, but we also have, together, lived through amazing experiences that made me become the person I am today.

To my roommates for the past year, Raúl and Dani, Dani and Raúl, for making even the most mundane days a experience worth remembering.

To my parents, who without ever asking for anything in return, have always believed in me and cleared the path for me so that I could reach every one of my goals . To my grandparents and rest of my family, for being always proud of me no matter what I did.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Speech processing and the use of technology for its analysis is growing at an exponential rate in todays society. Cooperation of machines and humans in building tools can change people's life. When this cooperation focuses on the machine learning about people's emotions and their changes it enables the creation of applications that less than an decade ago were seen as unreal.

1.1. Motivation

Emotions are affective states that human beings feel all the time. These states are subjective to the environment and come with physiological changes. They can also change depending on the subject's background since external signals do not affect in the same way to people with different life goals, priorities and life experiences.

Stimulus	Cognition	Emotion	Behavior	Effect
threat	"danger"	fear	escape	safety
obstacle	"enemy"	anger	attack	destroy obstacle
gain of valued object	"possess"	happiness	retain or repeat	gain resources
loss of valued object	"abandonment"	sadness	cry	reattach to lost object
member of one's group	"friend"	acceptance	groom	mutual support
unpalatable object	"poison"	disgust	vomit	eject poison
new territory	"examine"	expectation	map	knowledge of territory
unexpected event	"What is it?"	surprise	stop	gain time to orient

Table 1.1. ASSOCIATION OF EMOTIONS WITH ACTION PATTERNS

Emotions appear when something relevant happens, but the importance given to each event is subjective and therefore it is not possible to accurately predict how someone is feeling after the event happening. For example, if someone is watching television and hears a noise, his reaction and the chain of emotions he feels depend on how he receives the stimulus, the degree of interest he has on what he is watching and many other factors which are unknown at first sight. However, emotions and the reactions that they cause

on human beings have been crutial for their survival and evolution in history. In the early years of our species, hearing or seeing a tiger that would produce fear on our body and make us run away is a clear example of an evolutionary advantage to the species preservation [1]. In table 1.1 can be seen the different effects stimulus have on humans.

Despite the importance that has always been given to human's rationality, i.e, studies, professional experience, etc... the so called "soft skills" have gained massive importance over these last few years. These soft skills are related to communication, courtesy, flexibility, responsibility and the ability to work in teams. Therefore, technical knowledge and other skills are nowadays worthless if they come with a lack of sense of humor, enthusiasm or any other skill related to emotional intelligence [2].

Over the past years, new technologies and their interaction with our life have taken over our everyday life. Actually, the emerge of new algorithms, applications and tools that seek for making people life easier is growing exponentially [3], [4]. In terms of emotions, thanks to machine learning is possible to learn about a person's internal state and detect changes in the levels of stress that are being experienced from the person's voice and facial expressions. Research on emotion recognition is making good progress and needs stay in the same track, not only for the develop of new every-day use applications but also extend emotion recognition to other matters such as investigation of car accidents, fight against terrorism, education, medicine or neuroscience.

The demand for effective emotion recognition systems keeps rising as research on the field and affective neuroscience keeps making progress that helps people on their personal life as-well aswell as their professional life. A big amount of professors and scientists of the field agree on that research has to be focused on the emotional signals produced by face, body and speech. A solution to this increasing demand is the development of a emotion recognition system whose input is speech.

1.2. Aim of the thesis

The main goal of this thesis is to develop a dimensional and discrete emotion recognition system. Two different approaches for the system are presented and compared.

The first approach consists on a two-level sequential system. In this approach, the first sub-system extracts high dimensional characteristics from the audio file containing the acoustic signal and from the audio transcription containing the linguistic information. These characteristics are then mapped, reducing dimensionality significantly, into a three dimensional space in which emotional states are represented. Afterwards, a second sub-system maps the three dimensional space into a number of sub spaces that represent discrete emotions hence, output points (x, y, z) from the first sub system can be mapped to the discrete emotions they represent using the subspace in which the point is located. This approach is based on the psychological theory of emotions developed and followed since 1977 [5] by J. A. Russell and A. Mehrabian.

The second approach is very similar to the first one as it also extracts the features from voice signal and speech transcriptions to map them into 3 dimensions. However, in this approach, the classification of the emotion into the discrete space is done in parallel to the regression of the 3 dimensional variables. This approach challenges the validity of the emotional model presented in [5] and its informative capability in prediction tasks.

It is very important that the prediction is independent of the speaker. The implementation of both systems requires a previous study of human voice characteristics, text processing, predictors and classifiers that enable the recognition of emotions from the features obtained.

In addition, this thesis' research encompasses two different data sets and compares their performance on the task. Research is also done on which feature set for both the acoustic and linguistic features is best. This thesis also focuses on transfer learning and how it can be applied to this field.

1.3. Outline of the thesis

The thesis structure follows this presented structure:

- **Chapter 1:** Thesis main goals and motivations are described.
- **Chapter 2:** State of the art. An explanation is given on the needed theoretical basis about emotions, acoustic analysis, linguistics analysis and machine learning so than the proposed system is clearly understood by the reader.
- **Chapter 3:** The thesis proposed systems are presented. Thorough analysis and details are given about employed data sets, text processing, audio processing, extracted features and machine learning models implemented.
- **Chapter 4:** Ran experiments are explained and their results are presented. Furthermore, reasoning about the obtained results is included trying to give theoretical explanations on the improvements the system obtains with each experiment.
- **Chapter 5:** This chapter introduces the final conclusions of the thesis and introduces possible paths for future research.
- **Chapter 6:** Socioeconomic and regulation are discussed. Social and economical issues related with the development of the thesis are explained.

2. STATE OF THE ART

The goal of this chapter is to introduce the reader to the field of emotion recognition and present the required concepts to fully understand the development of this thesis. An explanation is given about the basis of emotions, voice characteristics, text processing and machine learning with its evolution until today.

2.1. Introduction

Audio files are one of the most used file formats nowadays. The signal produced by the voice is main used channel for communication among humans, and it is considered to carry a huge amount of information about the emotional state of the transmitter [6]. However, emotion recognition research has been mostly focused on facial expressions due to the technical difficulties that come with the use of audio. Although Charles Darwin, in 1872, stated the importance of speech in emotion recogntion, it wasn't until last century's 70s that in-depth research was conducted on speech.

Research on speech's role in emotions is split into two main groups:

- **Expression's research:** This research focuses on determining how an emotional state is expressed trough speech. The first works on this path of research were from Fairbanks and Pronovost [7]; and Williams and Stevens [8].
- **Recognition research:** This type focuses on finding how good is the estimation the receiver makes, trough acoustic-only voice characteristics, of the emotional state of the transmitter. The biggest problem that was faced by researches before the emerge of artificial intelligence was the separation between linguistics and acoustics in the voice channel. Linguistic channel transmits the content of the message, whereas the acoustics transmits characteristics such as pitch, volume or voice speed. Splitting both channels into independent ones achieves that the classification of an emotion is independent of the message contained in utterance. The first works in this subject were conducted by Soskin and Kaufmann (1964), in which frequency filtering was employed; and Fairbanks and Pronovost (1939) [7], in which the same phrase was employed for all emotions.

Current research on emotion recognition is based on machine learning algorithms, audio feature extraction and development of data sets oriented to emotion recognition. Therefore, research on the automatic recognition of emotions, whether it is using facial expressions or voice, does not only englobe the field of machine learning but also the fields of psychology, linguistics and biology. In psychology, emotions are syndroms generated from eventful situations; in linguistics, changes in voice frequency, spectrum or duration

of intensity are what lead to emotions; in biology, emotions are evolutionary patterns that body internal systems follow due configurations set up by the body in order to effectively deal with specific situations. Therefore, in order to make progress in emotion recognition systems, is necessary to make progress in the mentioned fields [9].

2.2. Emotional Foundation

This thesis aim is not to explain the entrails of an emotion. Nevertheless, it is important to understand them in order to shape how emotional states are constructed and it helps to discover which characteristics and relationships are useful to describe each emotional state. Most people have a vague and informal definition of emotion, but it is convenient to have into account the ideas from important personalities in philosophy, René Descartes, and mainly in psychology, William James. Descartes (1644) introduced the idea that only a few basic emotions shape the emotional life of human beings, whereas James (1884) in his thesis presented that an emotion is a feeling resulted from body changes in reaction to an external event, hence an emotion is not only what we feel [10]. A hundred years later, Izard (1994), a North American psychology researcher from the University of Delaware, stated that to correctly define an emotion it is necessary to take into account all the internal processes that happen inside our brains and nervous system, and the observable expression of each person. As it can be seen, emotion is a vague and imprecise idea due to the amount of studies about it and the different perceptions from each one of us.

The word emotion comes from the Latin word *emotio*, noun derived from the verb *movere*, which means "moving", since an emotion releases a person from its usual state and appears unexpectedly. There are three main components that define emotions:

- **Neurovegetative component:** These are physical reactions of our bodies that are controlled by the nervous system. For example, if someone is afraid, the nervous system makes our heart pump faster and lowers our voice pitch to a monotonous one.
- **Behavioral component:** When someone experiments an emotion, his or her behavior changes and it can be inferred from the changes in the pitch of his voice or the different facial expressions. This behavior allows other to know the mood of the person feeling the emotion and helps in building social relationships and empathy.
- **Cognitive component:** These are the feelings that an emotion produces in an individual. It is a subjective component, and tells us how each person handles each emotion [11].

Many researchers agree on that emotions and their interpretation are biased by culture, expectations, rules and environment that each person is currently experiencing. Different cultures treat differently emotions such as love, anger or shame.

In 1972, the North Americam physiologist Paul Ekman, who was a pioneer in the study of emotions and their facial expression, went against this interpretation, and still nowadays big part of the scientific community questions Paul's research. Paul decided to travel to Papúa, Nueva Guinea in order to study the facial expressions from the members of the Fore tribe, which live separated from other cultures. The memebers of this tribe were able to determine the emotions of people in photos by their facial expression, and that is how Ekman concluded that there existed 6 basic and universal emotions: happiness, disgust, fear, surprise and sadness. In the 90s, Ekman added a list of secondary emotions which are build up from two or more basic emotions, and that are not always easily identified from facial expressions, table 2.1 [12]. In addition, the psychologist and professor of the University of South Florida, Robert Plutchik, created in 1980 his own list of basic and secondary emotions, known as the "Wheel of Emotions" that represents the combination of basic and secondary emotions as it can be seen in figure 2.1. These models of emotions are known as discrete models.

Primary Emotions	**Secondary Emotions**
Happiness	Relief
Disgust	Embarrassment
Anger	Guilt
Fear	Contempt
Surprise	Amusement
Sadness	Excitement
	Pride
	Pleasure
	Shame

Table 2.1. PRIMARY AND SECONDARY EMOTIONS, PAUL EKMAN

Fig. 2.1. Plutchik's wheel of emotions

However in 2005, Cohen stated that the framework of basic emotions did not have any empirical evidence to support it. Furthermore, Cohen claimed that automatic responses and pan-cultural facial expressions did not provide a basis to think that the existence of a set of basic emotions was realistic [13]. The discrete model of emotions allows to represent a constraint set of emotional states whereas the dimensional model provides a wider range of emotional states to represent. The dimensional model takes into account two or three variables in order to represent the affective state in a multi-dimensional space [13]. Russell's circumflex model is an early model, in which an affective state is viewed ass a circle in the two-dimensional bipolar space [14], valence and arousal (Figure 2.2). Valence represents how positive an emotions is, values of valence close to zero mean neutral emotions. Arousal indicates how strong an emotion is felt, a high value of arousal means the emotion is having an strong impact whereas values close to zero mean we are feeling that emotion but is not really impacting us heavily. Using this two-dimensional biopo-

lar space 28 affective states can be represented: happy, delighted, excited, astonished, aroused, tense, alarmed, angry, afraid, annoyed, distressed, frustrated, miserable, sad, gloomy, depressed, bored, droopy, tired, sleepy, calm, relaxed, satisfied, at ease, content, serene, glad, and pleased.

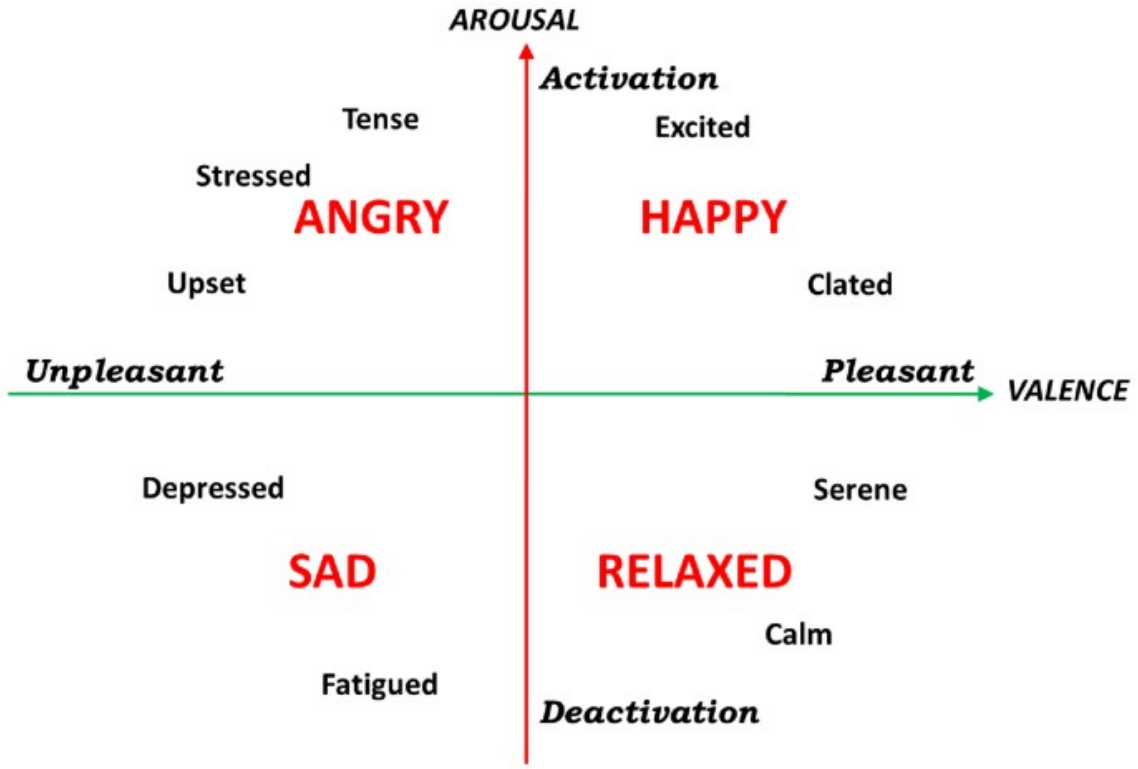


Fig. 2.2. Russell's circumflex model

Mehrabian and Russel in [5], [15] introduced a three-dimensional model, called the valence-arousal-dominance (VAD) in figure 2.3. In VAD valence and arousal mean the same as in the two-dimensional model. Dominance is the new variable added and this variable reflects the level of control of the emotional state, from submissive to dominant.

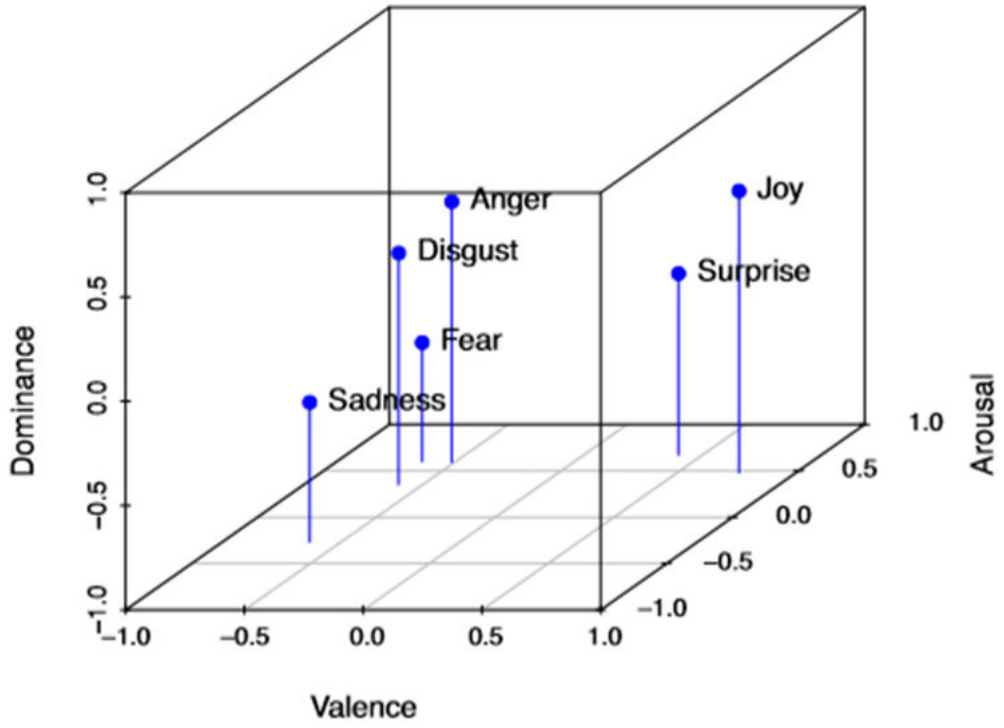


Fig. 2.3. Mehrabian and Russel VAD model

2.3. Speech Analysis

Speech is the sound produced by humans when air leaves the lungs trough the larynx and makes the vocal cords vibrate. Its emergence comes from the necessity of human beings to communicate, however, its use is not limited to communication purposes since it not only transmits messages but it also conveys emotional information.

2.3.1. Speech Paralinguistics

Voice model

Sounds emitted by humans can be tonal, produced by the relaxed oscillations of vocal cords like in consonants and vowels, or not tonal, if during the transmission vocal cords remain opened and air coming out produces turbulence. When producing tonal sounds, the flux of air coming out the lungs increments the pressure and vocal cords are expanded away, allowing air to flow. This flux of air lowers the pressure making the vocal cords to contract back again. The cycle in which vocal cords are expanded and contracted is known as the glottic cycle.

In figure 2.4 is represented the system of production and synthesis of voice. The input signal goes through the vocal conduct (H[z]), where the signal is amplified and modified

by resonance frequencies where most of the sound energy is concentrated.

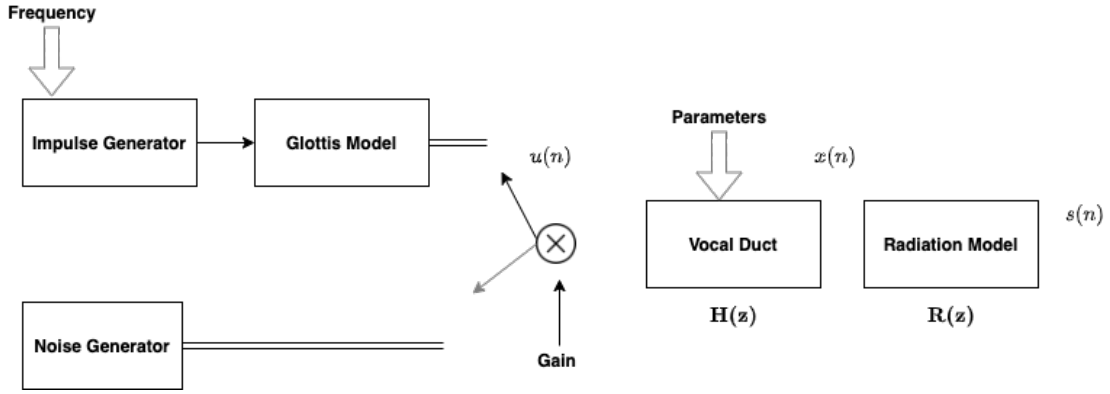


Fig. 2.4. Diagram of voice production model

The vocal conduct is represented through the transfer function in equation 2.1, and the radiaton model uses a high-pass filter. This is represented by the impedance R[z] produced by air's pressure when comming out of the lips, shown in equation 2.2.

$$H[z] = \frac{1}{1 + \sum_{k=1}^{p} a_k z^{-k}} \tag{2.1}$$

$$R[z] = \left(1 - \alpha z^{-1}\right) \tag{2.2}$$

Gain, G[z], models the glottis. However, in this case, in order to simplify the transfer function the glottis model nor the radiation model are considered. Equation 2.3 represents the transfer function of the whole system.

$$F[z] = G[z]H[z]R[z] = \frac{G}{1 + \sum_{k=1}^{p} a_k z^{-k}} \tag{2.3}$$

Finally, the voice signal is modeled as s[n], see equation 2.4, where u[n] represents the excitation. If sound is tonal, the excitation is white Gaussian noise, it it is not tonal, u[n] is modeled by equation 2.5

$$s[n] = Gu[n] - \sum_{k=1}^{P} a_k s[n-k] \tag{2.4}$$

$$u[n] = \frac{1}{N_0} \sum_{r} \delta\left[n - rN_0\right] \tag{2.5}$$

where N_0 represents the fundamental period, inverse to the fundamental frequency F_0, and a_k are the coefficients for the lineal prediction that must minimize the mean quadratic error function 2.6.

$$\epsilon[n] = \sum_{n=-\infty}^{\infty} (s[n] - \hat{s}[n])^2 = \sum_{n=-\infty}^{\infty} \left(s[n] + \sum_{k=1}^{p} a_k s[n-k] \right)^2 \tag{2.6}$$

Voice characteristics

Sound is propagated trough a sinusoidal wave since it is the result of the vibration in air's particles and like every other wave, it has physical characteristics: wave length, amplitude, frequency. Nevertheless, voice presents a set of characteristics which are dependent on the morphology of the organs in the speech system:

- **Intensity:** It proportional to the amplitude of the wave which represents the infraglottic pressure. This pressure depends on the space between vocal cords when these are opened. The bigger the infraglottic pressure, the higher will the intensity be. Following this evidence, a high intensity can be associated with stress and a low intensity can be relate with depression or tireness. The unit of measure for intensity is the decibel (dB).

- **Pitch:** It determined by the number of glottic cycles inside a time unit. A glottic cycle is detailed in the previous section. This characteristic depends on the vocal folds, hence the size of the pharynx, and it affects the voice's fundamental frequency. This is the reason why, a woman has a higher pitch than a man. The fundamental frequency shifts are the result of elastic changes that the vocal cords experiment. The unit of measure for pitch is Hertz (Hz).

- **Timbre:** It allows to distinguish two sounds with same pitch and intensity. It depends on the morphology of the resonant organs, hence each person has an identifying voice.

- **Duration:** It is determined by the volume and the velocity of air that is coming out. The bigger the lung capacity and the size of the rib cage are , the longer the duration is.

- **Loudness:** Depends on the force of the air that arrives to the larynx, the intesity and the infraglottic pressure.

Now that the main voice characteristics have been presented, a list is presented in which emotions are related with the voice characteristics that define them:

- **Anger:** *"In comparison to neutral speech, anger is produced with a lower pitch, higher intensity, more energy (500 Hz) across the vocalization, higher first formant (first sound produced) and faster attack times at voice onset (the start of speech)."* [16]

- **Disgust:** *"In comparison to neutral speech, disgust is produced with a lower, downward directed pitch, with energy (500 Hz), lower first formant, and fast attack times similar to anger. Less variation and shorter durations are also characteristics of disgust."*[16]

- **Fear:** *"Fear can be divided into two types: "panic" and "anxiety". In comparison to neutral speech, fearful emotions have a higher pitch, little variation, lower energy, and a faster speech rate with more pauses."*[16]
- **Sadness:** *"In comparison to neutral speech, sad emotions are produced with a higher pitch, less intensity but more vocal energy (2000 Hz), longer duration with more pauses, and a lower first formant."*[16]
- **Happiness:** High pitch and intensity with increments in the speech speed.
- **Surprise:** Medium pitch and intensity, slightly over the average intensity. Speech speed does not change

2.3.2. Speech Linguistics

Language's smallest meaningful representations are words. Words are crucial in the understanding and description of the world that society has. Actually, the principle of linguistic relativity, known as the SapirWhorf hypothesis, states that the structure of a language has an impact on how the speakers of that language think [17].

The are some words such as: good, bad, terrible, excellent, nice and more that denote valence since valence is the their core meaning [18]. On the other hand, there are other words that have strong or negative connotations without explicitly denoting valence. In the case of party or raise, these words have a positive valence connotation, on the contrary, words like slave and death have a negative valence connotation. In the middle between these two extremes, neutral words exist, which are neither associated with positive or negative valence. It is hard to put boundaries on when a word becomes neutral, positive or negative since it depends a lot on the receiver. Nevertheless, for the most common used terms in a language there is high inter-speaker agreement on the connotation of such words. In addition, there are words whose core meaning is expressing an actual emotion, hence they are associated with that emotion. However, some words' core meaning is not an emotion but they still have an association with an emotion. For example, the words rage and anger explicitly evoke anger, and they are associated with anger, but in the case of betrayal or negligence they do not explicitly state anger and they are associated with anger. Emotion association lexicons capture how phrases and words are related to valence and emotions.

- **Delighted** - Associated with JOY and POSITIVE valence
- **Death** - Associated with SADNESS and NEGATIVE valence
- **Shout** - Associated with ANGER and NEGATIVE valence
- **Furniture** - It is not strongly associated with neither a high/low valence value or with an emotion

However, several challenges arise when trying to automatically detect a sentiment from a word or phrase:

- The emotion transmitted by a phrase is not just the sum of the emotional meanings of the words that make up a phrase. The relation between words and the position of words matter an can influence the meaning of the sentence as a whole. Furthermore, emotions are not often said explicitly[18]. For example: *"Another Tuesday, and another week that I am working my ass off"*. This sentence without using overly negative words or explicitly saying so, it conveys a sense of frustration.

- There are some terms, like modals or negations that have an impact on the emotional meaning of a sentence. These terms when by themselves do not have a strong connotation but in conjuction with other terms they modify the meaning, i.e, was bad, may be bad and was not bad do not mean the same.

- Many times a word or phrase can have multiple emotion and valence associations depending on the situation [18]. For example the word *hug*. Depending on the context this word is used it represents a different connotation. In the phrase: *"Anotonio hugged his child before leaving the house"*,*hugged* is associated with joy and affection. In contrast, when we say *"The pipeline hugged the state border"*, *hugged* relates to staying close and has an unemotional connotation.

In addition, when trying to build a model to detect an emotion from a word or a phrase, it is very important to think about the different languages. Languages impose a big barrier for the model since each idea, object or concept from the world is represented by a different word in each of the existing languages. Therefore, if the model is trained on a specific language, it will not be able to extract the emotional meaning from a word of a different language than the learned one even-though both words represent the same in real life. A possible way of going around this issue is to translate words to the learned language, but current translation methods do not always reference to the correct word and are sometimes misleading. In the case that an ideal translation method existed, and it always translated a word to its correct representation in other language, there would still be a barrier for the model. This barrier is culture, languages are are often related to the culture of the people that speak it and to geographical traditions. In this way, some ideas or concepts might have a positive connotation for one culture and a negative connotation for a different culture, making the model have to distinguish cultures which is out of the scope of this thesis. On top of that, some languages have different balances of words representing an emotion. In the case of English, it has 252 words expressing disgust that represent 22% of the English WordNet-affect vocabulary and Spanish has 206 words for disgust that represent 5% of the Spanish vocabulary [19]. These imbalances make it hard to evenly learn emotion association for words since the vocabularies of languages have such imbalanced emotions.

Emotion	Number of words
Happiness	1,177 (32.7%)
Disgust	206 (5.9%)
Anger	959 (26.7%)
Fear	431 (12%)
Surprise	267 (7.4%)
Sadness	551 (15.3%)
Total	3,591

Table 2.2. SPANISH VOCABULARY EMOTIONAL ASSOCIATIONS

Emotion	Number of words
Happiness	399 (35.9%)
Disgust	252 (23%)
Anger	137 (12.3%)
Fear	53 (4.7%)
Surprise	67 (6%)
Sadness	202 (18.1%)
Total	1,110

Table 2.3. ENGLISH VOCABULARY EMOTIONAL ASSOCIATIONS

2.4. Machine Learning

The following sections present a review of history up until today of machine learning and artificial intelligence and explanations of concepts of the field used later in the thesis.

Artificial intelligence is the broader term to make reference to the field of computer science that is centered in creating computer programs that have an intelligent behavior. It was minted in 1956 by the computer scientist John McCarthy in the conference at Dartmouth [20]. It encloses multiple methods, such as search algorithms, statistics, predictive analysis and machine learning.

More in depth, machine learning is technique based in the use of algorithms to analyze data, learn from data and use the data for prediction. Furthermore, machine learning also encloses what is known as deep learning. This type of learning focuses on algorithms that try to mock how the human brain works in order to process data, generate patterns and make decisions from a set of data.

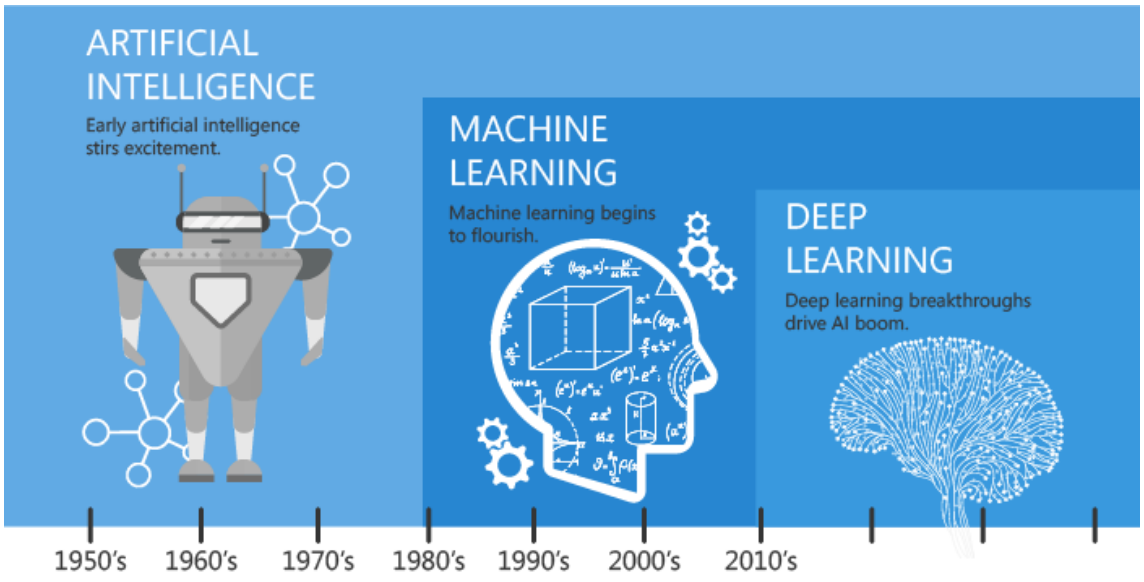


Fig. 2.5. History of the field

2.4.1. History

Machine learning started by the 50s of the past century by the hand of Arthur Samuel. Arthur was a pioneer in the field of computer games and artificial intelligence who designed a checkers game in which the more games the computer played, the better it played the game. To achieve this, Samuel studied the winning moves in the games between humans and he added them to the computer. In the same decade, 1957, Frank Rosenblatt invented the perceptron, this technology imitates neuron from a human brain.

In 1960, the algorithm known as "nearest neighbor" to recognize patterns was invented by the hand of Michael F. Dacey. The original concept of this algorithm is that all given points are randomly distributed in space following a Poisson distribution. In 1970, a student group from the University of Stanford coded the *Stanford Cart*, a self-driving golf cart that was able to function by sending pictures to a computer through a wireless connection. This receiver computer analyzed the pictures and decided where the cart had to move to, sending back to the cart the decision with the moving commands [21].

In the 90s, computer science and statistics came together to give a probabilistic look of artificial intelligence problems. From that moment on, algorithms started to work with more data, and they started to be applied on language translation, data mining and web applications. In addition, Geoffrey Hilton introduced the concept of neural networks that allows computers to distinguish objects and text in images.

Finally, since 2010 until today, big tech companies have been developing their own tolls for machine learning. Some examples are Google and Stanford University with *GoogleBrain*, IBM with *Watson* or Microsoft with *Kinect*.

2.4.2. Types of Learning

It is important to know about the different types of learning that machines can implement since this allows us to have an intuition of which one to use when facing a new problem and understanding why and how the chosen approach works. The main types of learning are:

- **Supervised learning:** This is the type of learning used in this thesis and the most popular one since its implementation is the simplest of all. With a set of labeled data, it is possible to feed an algorithm so that it learns to approximate the function that relates each sample in the data set with its label.

$$y = f(x) \tag{2.7}$$

  teh function $f(x)$ transforms the input **x** in the output variables, **y**. While training the algorithm, the parameters of function $f(x)$ are optimized, and when training is done, it will be able to predict a correct label from an unseen and unlabeled sample. This type of learning is split in two main approaches:

  - **Classification:** When the output variable is categorical or qualitative value, such the prediction of a malignant or benign tumor in terms of its characteristics or the sport being practiced in terms of the movements made by the person.
  - **Regression:** When the output variable is a quantitative value, the price of a house in terms of the squared feet, location and number of rooms.

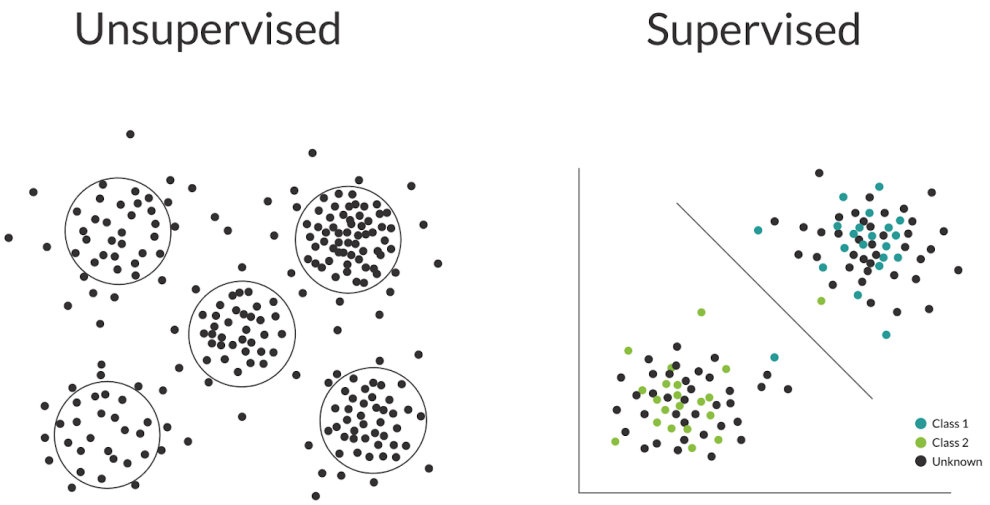


Fig. 2.6. Supervised vs Unsupervised Learning

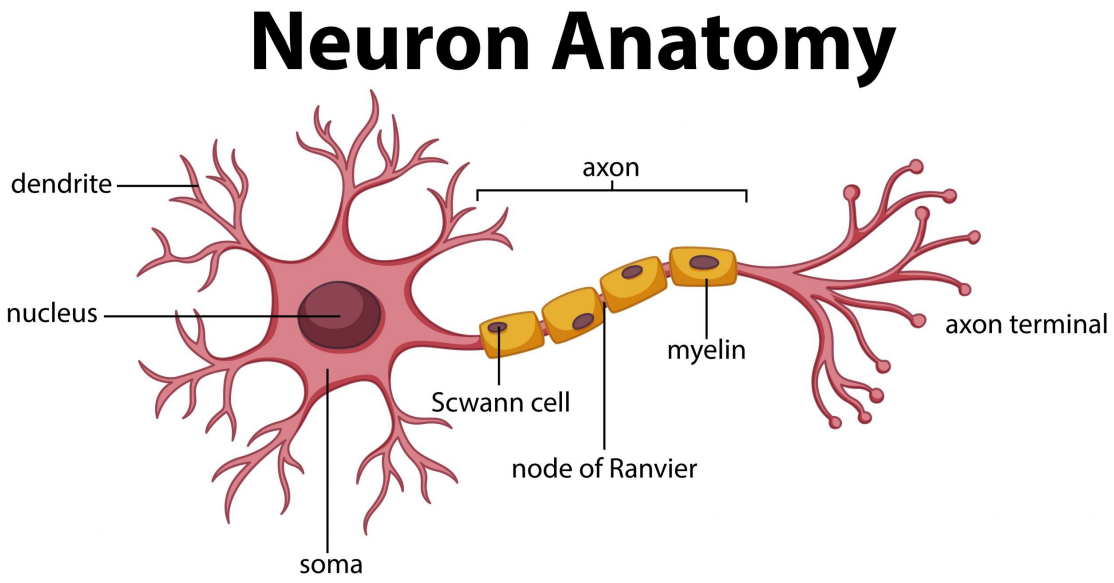
- **Unsupervised learning:** In this type of learning, instead of having labeled data, an unlabeled data is given in order to study the structure of the data and its distribution. Afterwards, data is organized in such a way that an algorithm or human is able to interpret such organization. This type of learning is extremely useful since most of real world data is unstructured and unlabeled, and algorithm that is able to make sense out of big amounts of data is very beneficial in many industries. Some examples of this type of learning are recommender systems like the one employed by Netflix in order to recommend content to users based on the likes of similar users; or online commerce that group their clients in terms of their shopping habits.

- **Reinforcement learning:** The main difference between supervised and unsupervised learning is very clear, and is whether or not labels are present. However, in the case of reinforcement learning, the differences are less clearer. Learning in this case, is focused on an error-based system, therefore, algorithms using reinforcement learning will fail a lot in the early stages of learning. After some iterations, in which "rewards" are given to the algorithm in case it behaves correctly and punishments when it misbehaves, the algorithm learns to make less mistakes and lowers its error probability. Some examples of this type of learning are video games and robots [22].

2.4.3. Proposed Models

Deep Learning: Feed Forward Neural Networks

Neural networks are a very important field inside the study of artificial intelligence. Human brains are complex systems in which an estimated number of 100,000 million neurons have 500 billion connections among them. Neurons are simple information processing units whose main components are the cellular nucleus, dendrites (input channels) and the axon (output channel). A neuron is able to receive up to 10,000 inputs and send its output to hundreds of other neurons. The connections among neurons is known as synapses and its performed through nervous impulses.



Artificial neural networks are based on biological neural networks. Similarly to biological neural networks, artificial neural networks are characterized by a set of inputs $[x_1, ..., x_n]$, synaptic weights $[w_1, ..., w_n]$ corresponding to each input, an output y and an activation function f. The output of the neuron is presented in equation 2.8

$$y = f\left(\sum_{i=1}^{n} w_i x_i\right) + w_0 \tag{2.8}$$

The main advantages that artificial neural networks provide us with are their ability to learn from examples, their speed in comparison with other methods and their noise tolerance. Nevertheless, they also have some disadvantages such as the high complexity of training, the trouble to correctly design the number of layers and the number of neurons per layer, the trouble to interpret results and their lack of scalability.

In 1969, the scientists Marvin Minsky and Seymour Papert showed that the simple perceptron (single layer neural network) and the Adaline (perceptron with an added linear adaptive mixer) were not able to solve non linear problems. In the wake of this finding, George Cybenko in 1989 showed that the multi-layer perceptron or MLP is a universal approximation method, meaning that it is able to approximate any continuous function.

An MLP is a type of artificial neural network used for both classification and regression that combines neurons in the input layer with neurons in the output layer using one or more layers of neurons in between. The dimensions of the input layer are the same as number of features of a sample in the data set, the dimensions of the output layer are the same as the number of categories or the number of variables to do regression on. The neurons in the "hidden" or middle layers have as input dimension the output dimensions of the previous layer.

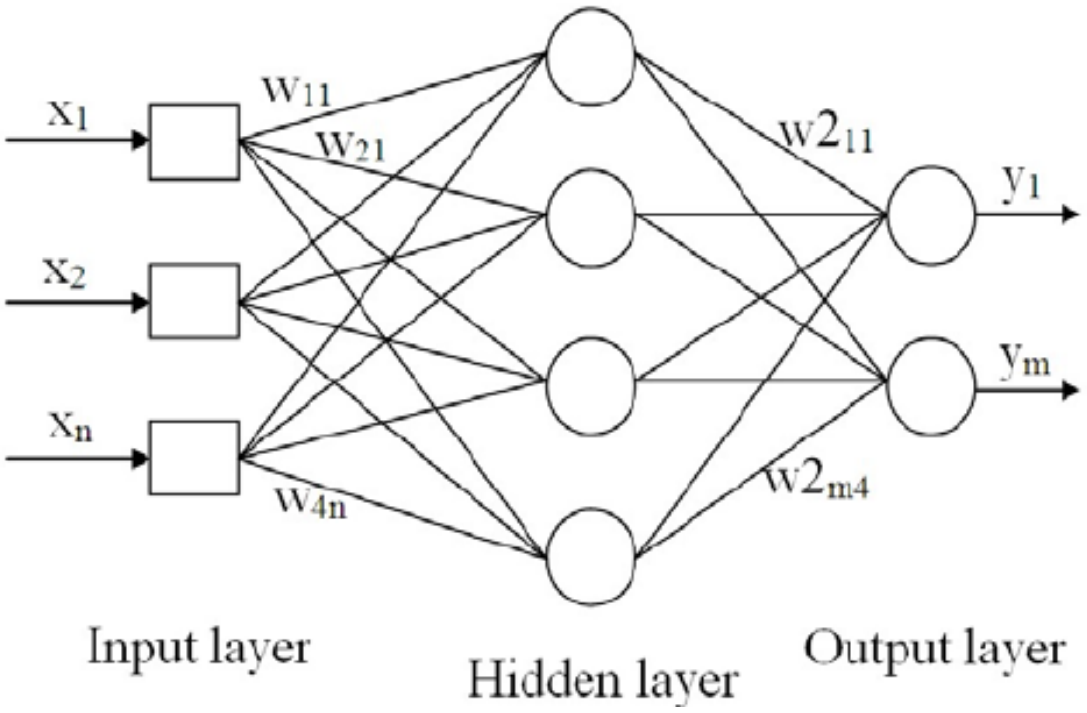


Fig. 2.7. MLP architecture

In the case of the classification of emotions, where there are 6 different emotions, the output vector y when emotion 3, happiness, is the correct and predicted output looks like this:

$$y = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{2.9}$$

And in the case of the regression of VAD values, where there are 3 output variables, the output vector of the network for $valence = -0.25, arousal = 0.5 and dominance = 0.12$ is the following:

$$y = \begin{bmatrix} -0.25 \\ 0.5 \\ 0.12 \end{bmatrix} \tag{2.10}$$

The choice of the activation function to use depends on the type of problem to solve by the neurons. Usually, the range of output values oscillates in the range $[0, 1]$ or $[-1, 1]$. One of the most used activation functions is the sigmoid function, equation 2.11, which transforms values into the $[0, 1]$ range where very low values map to 0 and higher ones to 1. Another commonly used is the hyperbolic tangent, equation 2.12, similarly to the sigmoid function this function maps values to a range but this time the range is $[-1, 1]$

where low values map to -1 and high values to 1. The disadvantage of both of this functions is that they have very slow convergence and do not cancel out negative number. The solution to this issue, is using the ReLU (Rectified Linear Unit), shown in equation 2.13, that cancels out negative values and keeps the positive values the same. The ReLU is not bounded and many neurons can vanish if there are a lot of negative values, however, it has a very good performance on audio and images, hence is the used function in this work [23].

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.11}$$

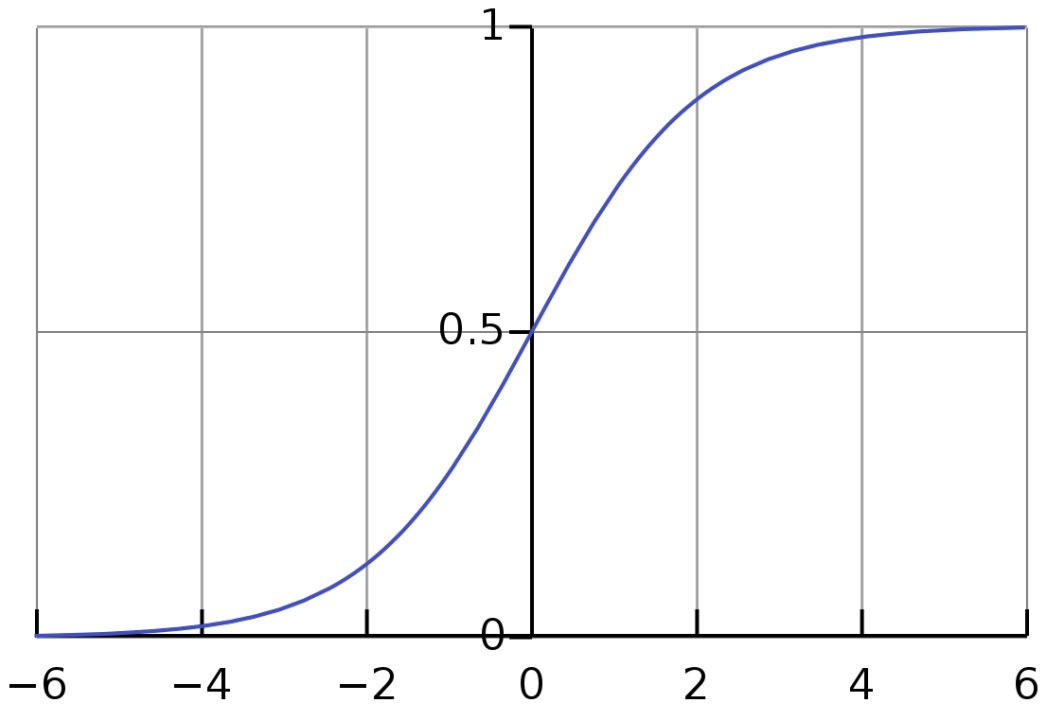


Fig. 2.8. Sigmoid function

$$f(x) = \frac{2}{1 + e^{-2x}} - 1 \tag{2.12}$$

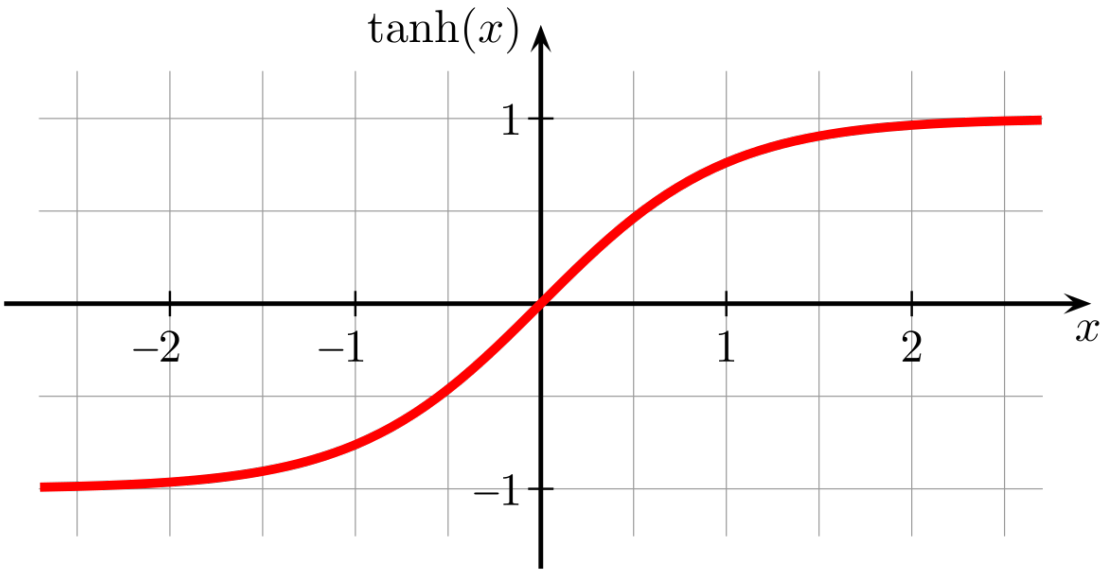


Fig. 2.9. Hyperbolic tangent function

$$f(x) = \begin{cases} 0 & \text{si} \quad x < 0 \\ x & \text{si} \quad x \geq 0 \end{cases} \tag{2.13}$$

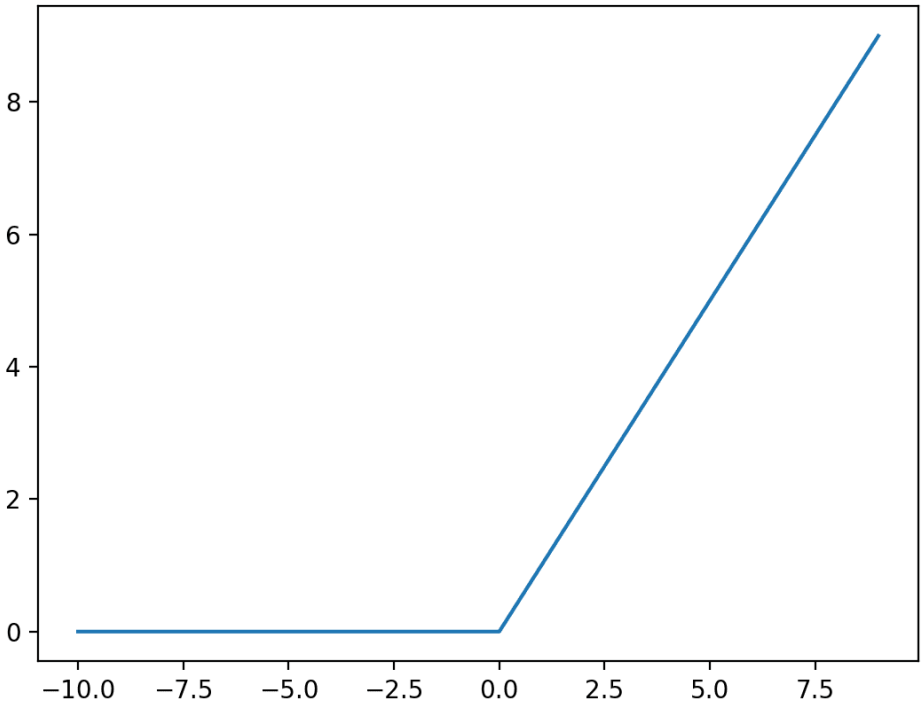


Fig. 2.10. ReLU function

Once the number of hidden layers, the number of neurons per layer and the activation are defined, the only left parameters to tweak are the weights in each neuron w_i. Therefore, given a data set with labeled samples, the goal is to find the weights that make the network predict a correct output when given an unlabeled sample. The method used to calculated the weights in training is known as back-propagation, since it propagates weights from the output layer to the input layer. Given an input vector $\mathbf{x}(n) = [x_1, x_2, ..., x_n]$ and an output vector $\mathbf{o}$(n), the goal is to find the weights such that $\mathbf{o}(n) = \hat{\mathbf{y}}(n)$, which minimizes the error function E, the mean quadratic error of all patterns:

$$E = \frac{1}{N}\sum_{n=1}^{N} e(n) \tag{2.14}$$

$$e(n) = \frac{1}{2}\sum_{i=1}^{n_c} (o_i(n) - \widehat{y_i}(n))^2 \tag{2.15}$$

where N is the number of patterns and for each patter $i, o_i(n)$ the desired output is $\hat{y}_i(n)$. The n_c parameters represents the number of neurons in layer c. In order to minimize the error function, the weights are optimized, i.e, in each iteration $e(n)$ is updated through the gradient:

$$w(n) = w(n-1) - \alpha\frac{\partial e(n)}{\partial w} \tag{2.16}$$

where α is the learning rate. In this fashion, weights are updated following the negative direction of the gradient of the error function. This is why the algorithm is known as "gradient descent".

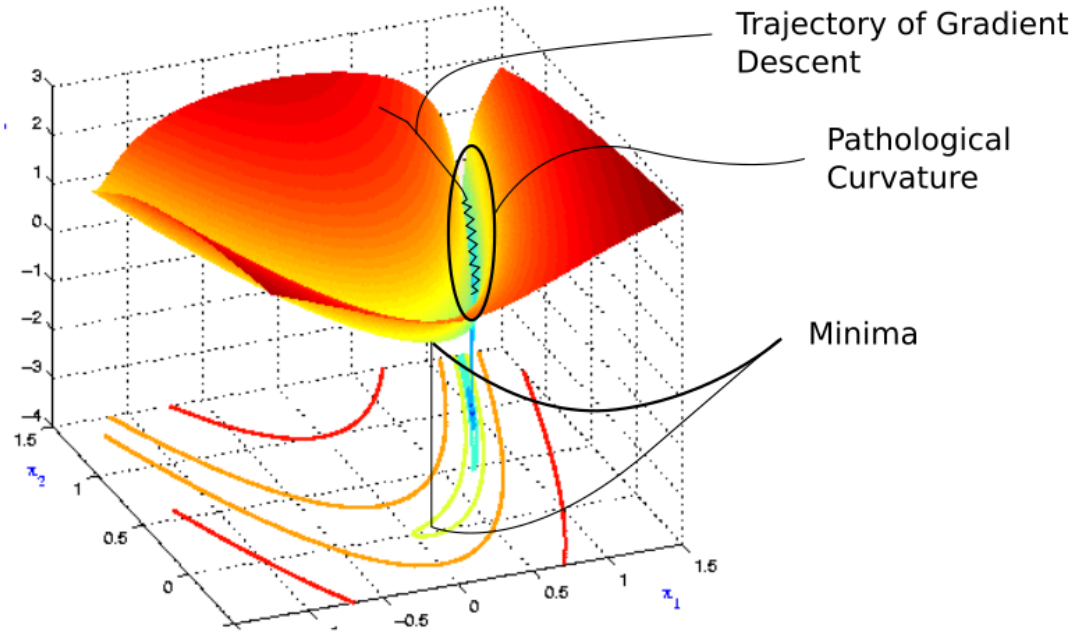


Fig. 2.11. Visual view of the gradient descent algorithm where the x, y axis are the weight vectors and the z axis is the value of the error

An important factor previous to the training of a neural network is the initial value of the weights. If weights are just simply initialized to 0, the derivative of the loss function $\frac{\partial e(n)}{\partial w}$ in each iteration is the same for all weights w_i in $\mathbf{w}$, hence all weights take the same value in each iteration, leaving the model with no learning at all. Therefore, it is very common to randomly initialize weights. The criterion to follow depends on the number of layers and the neurons per layer but is convenient that weights are close to 0. In this thesis, the Xavier initialization method is used [24]. Xavier method initialized method following a normal distribution in the range $[-l, l$ where:

$$l = \sqrt{\frac{6}{a-b}} \tag{2.17}$$

where a is the number of units in the input and b is the number of units in the output [24].

Finally, it is crucial to mention the importance of feature normalization of the data to train and evaluate a MLP. This process of normalization prevents the input features from having different impacts on the network. For example, the first MFCC represents the energy of the signal and ranges between very large negative values $(-200, -500)$.However,

the second coefficient indicates the global energy between low and high frequencies which can range from 0 to 100. This difference in some characteristics makes that some are more influence on the network than others leading to possible overfitting in the network and wrong decisions. To avoid this behavior all features are normalized to be in the same scale with mean 0 and variance 1. **Deep Learning: Recurrent Neural Networks and LSTMs**

Recurrent neural networks (RNN) were first shaped in the 1980s. However, these networks have been very difficult to train due to their computational complexity, hence, it was not until a few years ago that they started to become popular and more accessible due to the progress made in computation power.

Up to know we have only presented networks whose activation function only works in the forward direction, from the input layer to the output layer, meaning that neurons do not remember previous values. A RNN is similar to the feed forward networks but adds a new connection that points backwards and provides feedback among the neurons inside the same layer. In each time step, the neuron receives as input the output from the previous layer and the output of the previous time step from the current layer. Now each recurrent neuron has two types of parameters, one for the input vectors from the previous layer and one for the input from the previous time step.

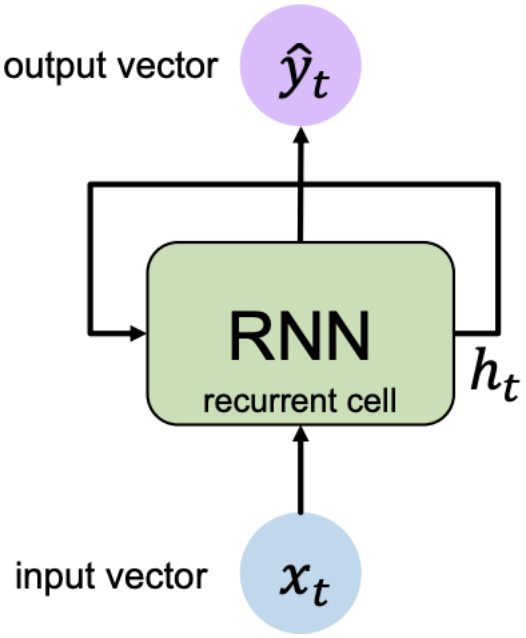


Fig. 2.12. RNN cell

$$\begin{aligned} h_t &= \tanh\left(\boldsymbol{W_{hh}} h_{t-1} + \boldsymbol{W_{xh}} x_t\right) \\ \hat{y}_t &= \mathbf{W}_{hy} h_t \end{aligned} \tag{2.18}$$

where $x = [x_1, x_2, ..., x_t$ represents the input sequence from the previous layer, W is the weights matrix and b is the bias. Now, in the back-propagation steps both weights from W_{hh} and W_{hx} are updated.

Since the output of a recurrent neuron at a time t depends on the input from the previous time step $t-1$, it could be said that a recurrent neuron has somewhat of memory. The part of a neural network that keeps an state throughout time is called *memory cell*. This memory feature is what makes these kinds of networks very suitable for learning problems where data is sequential.

In the previous section back propagation is presented as a method that traverses the neural network backwards in order to calculate the partial derivatives of the error function with respect to the neuron's weights. These derivatives are used by the gradient descent algorithm to iteratively minimize the loss function by optimizing the weights. However, in the case of RNN the method has some changes and its known as Back-propagation Trough Time (BPTT). If we unfold a layer of neurons it can be seen how information is passed from one time step to the next one. Analyzing this unfolded layer it is observed that a RNN can be considered to be a sequence of normal neural networks, one per time step with its own normal back-propagation. When applying BPTT, it required that the concept of unfolding is included mathematically, since the loss from a concrete time step depends on the previous one. Inside the BPTT, the error is propagated backwards from the last time step to the very first one. This allows to calculate the loss for each time step and hence update the corresponding weights. This non-cyclical graph results from the fact that the time unfolding is huge and hence computing BPTT has a huge computational complexity.

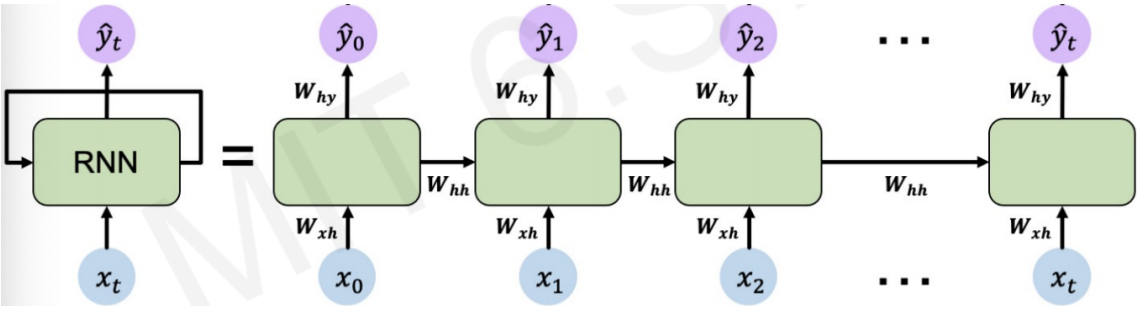


Fig. 2.13. RNN unfolded [25]

This complexity of the BPTT algorithm comes with two typical issues that affect all neural networks but specially recurrent ones, exploding and vanishing gradients. A gradient is a partial derivative of a function with respect to the function inputs that measures how much the function output changes when the input changes. It can also be interpreted as how much the function's slope changes at a point, the bigger the gradient, the steeper the slope and hence the faster the model learns. In summary, the gradient indicates how to change the weights in order to minimize the error.

Exploding gradients occur when the algorithm assigns an excessively high importance to weights, without any reason, leading to a problem in learning. This issue is easily fixed by reducing or clipping the gradients to a maximum value. On the other hand,

vanishing gradients happen when the values of the gradient are too low and hence the model stops learning or requires too much time to properly learn. This was a big issue during the 90s and was a lot harder to solve than exploding gradients. This issue was finally solved by using *gate units*.

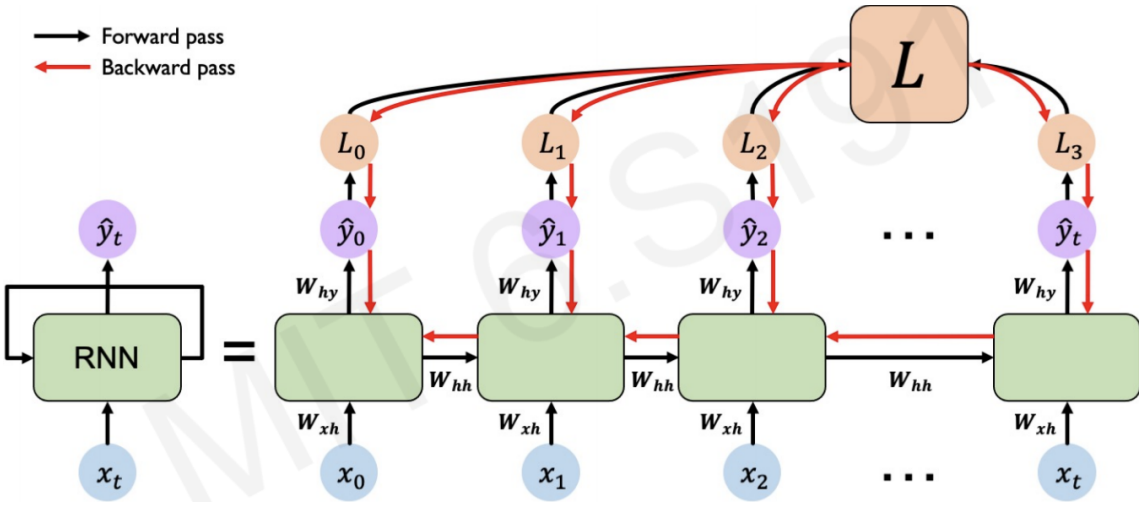


Fig. 2.14. BPTT graph where L is the loss for each time step. [25]

Long-Short Term Memory (LSTM) neural networks are a type of RNN that can retain information inside their memory for long periods of times. This is possible tanks to the fact that LSTMs memory works similar to a computer's memory, since LSTMs cells can write, read or delete information from its memory. This memory can be seen as a "blocked" cell, where blocked means that the cell decides whether to store or destroy the information inside (opening or closing the gate) depending on the importance of the new information arriving to the cell. Importance is assigned trough weights, which also learn during the algorithm. Therefore, this networks learn throughout time which information is relevant and which one is not. In a LSTM neuron, there are three gates to access this information: input gate, forget gate and output gate. This gates determine whether or not a new input is allowed, information is deleted because is no longer relevant or if it affects the current time step output.

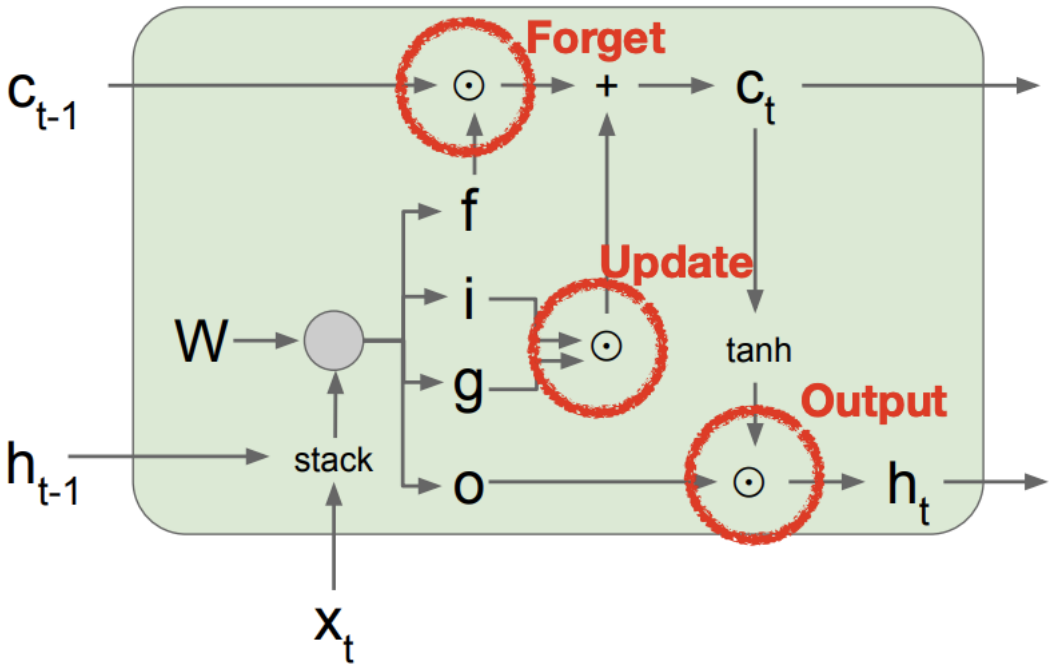


Fig. 2.15. LSTM cell.

$$
\begin{aligned}
f &= \sigma\left(W_{fx}x_t + W_{fh}h_{t-1} + b_f\right) \\
i &= \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + b_i\right) \\
o &= \sigma\left(W_{ox}x_t + W_{oh}h_{t-1} + b_o\right) \\
g &= \tanh\left(W_{cx}x_t + W_{ch}h_{t-1} + b_c\right) \\
c_t &= f \circ c_{t-1} + i \circ \tilde{c}_t \\
h_t &= o_t \circ \tanh\left(c_t\right)
\end{aligned}
\tag{2.19}
$$

These gates implement a sigmoid function which allows them to be included in the back-propagation process. Furthermore, these gates solve the vanishing gradient problem since they help the LSTM keep the gradients steep enough so that training is short and accuracy is high.

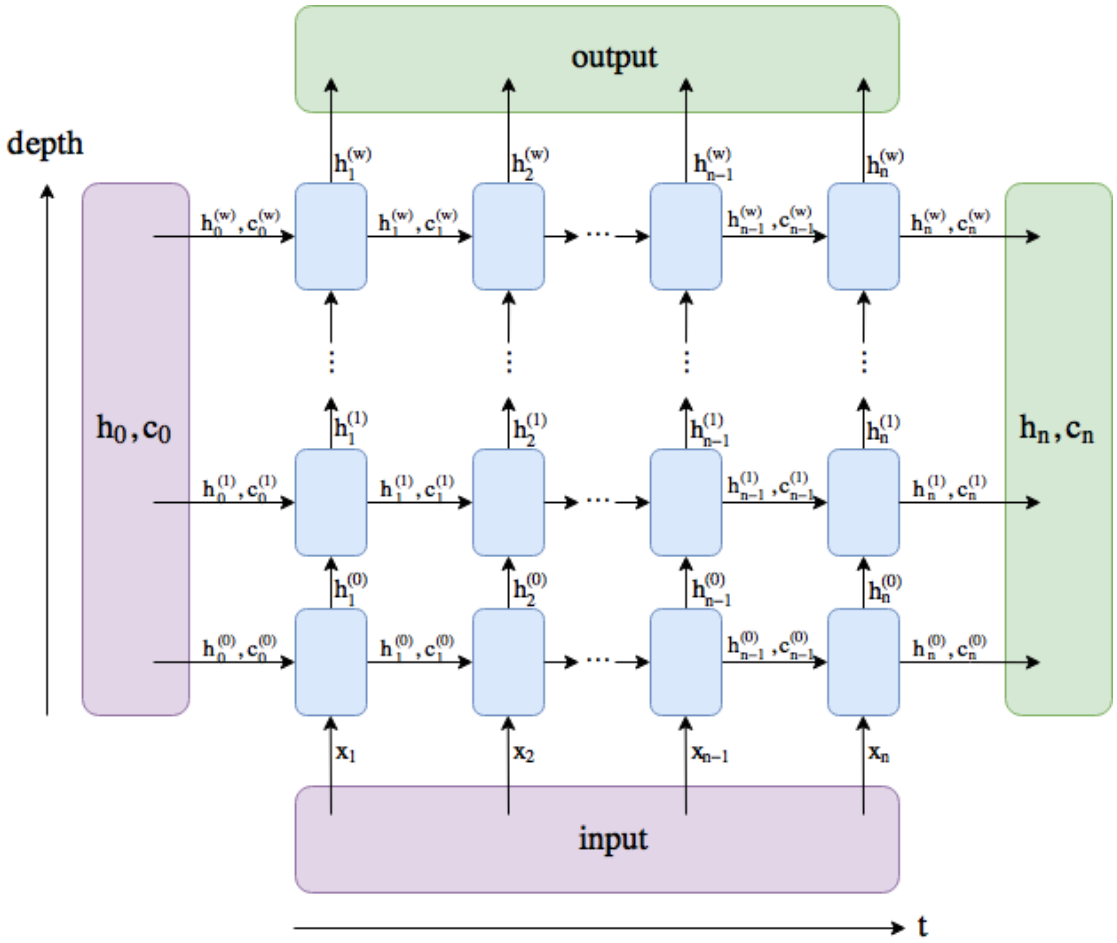


Fig. 2.16. LSTM graph.

Transfer Learning

Transfer learning is one of the most important techniques in Deep Learning. This method consists in taking advantage of the information learned by a model A in one task by having another model B on a different but similar task reuse the learning of model A.

The first step is to choose a general problem of a domain, for example, transcribing audio to text. Then a big and deep model is trained on such problem. Ideally this model, called pre-trained model, is big and deep enough so that it has to run for weeks until it converges. This pre-trained model has learned to transfer audio to text so it has acquired patterns to distinguish words in sentences, silence, human voices from background noises and many more, all of these skills gained by the pre-trained model are very general in the field. Therefore, when there is need to solve a more specific problem in the same field, we can reuse the learning from the pre-trained model on the new model. The newer model will acquire all the general-purpose skills obtained by the pre-trained model and then it will be "fine-tuned" for its specific task. Fine-tuning is the process of adapting the learning from a pre-trained general model for a more specific task, i.e changing the number of outputs in the last layer since the pre-trained model was outputting only a word

(1 output) for an audio and now we want the model to output 3 variables that describe an emotion.

At the end of the day, what it really means that "learning is transfered from the pre-trained model" is that instead of having the new model randomly initialize the weights and biases, the new model uses the ones from the pre-trained model as its starting point. Since this weights and biases are for a general-purpose task but similar to the model's task, they are a great starting point for the model and much better than random, leading to a better performance.

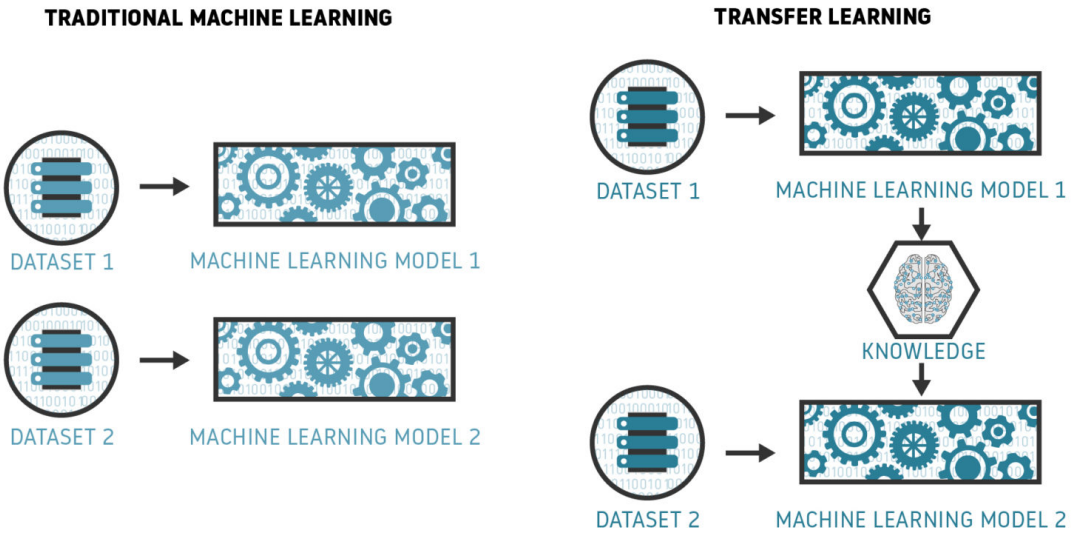


Fig. 2.17. Transfer learning vs traditional learning [26]

This concept is so powerful not only because it allows to get better performance of models in very specific task but also because of the computational cost reduction that it provides. As mentioned in the beginning, the pre-trained model is trained for weeks, hence it is very computationally expensive to train it and its usually trained by big institutions who have tons of resources. Therefore, if we did not have these pre-trained general-purporse models and we wanted to approach specific tasks we would have to do this heavy training for each of the models we develop and it most cases it would be impossible since most of us do not have the resources or the time to train it. This is one of the biggest advantages of transfer learning, enabling an individual to just load a incredibly large model which would be impossible for him to train and have the opportunity to use it as an starting point for any specific task of his research making the process last days or hours instead of months.

Random Forest

Random forest is a very useful useful algorithm for multi-class problems due to its great generalization capabilities. It is made up of a combination of decision trees in which

each tree depends on a random vector with the same distribution [27].

In order to understand how random forests operate is necessary to analyze how decision trees work since they are the building blocks of random forests. Decision trees learn from the data a set of rules to make decisions. The goal is to found the characteristics that allow to split the data set in smaller groups such that these groups are the most different among them. Following this, the data is split in smaller subgroups as the decision tree grows creating nodes and rules. A decision node is associated to a characteristic and can have two or more branches depending on the number of classes that is going to split. The leave nodes ideally belong to only one class as it can be seen in figure 2.18.

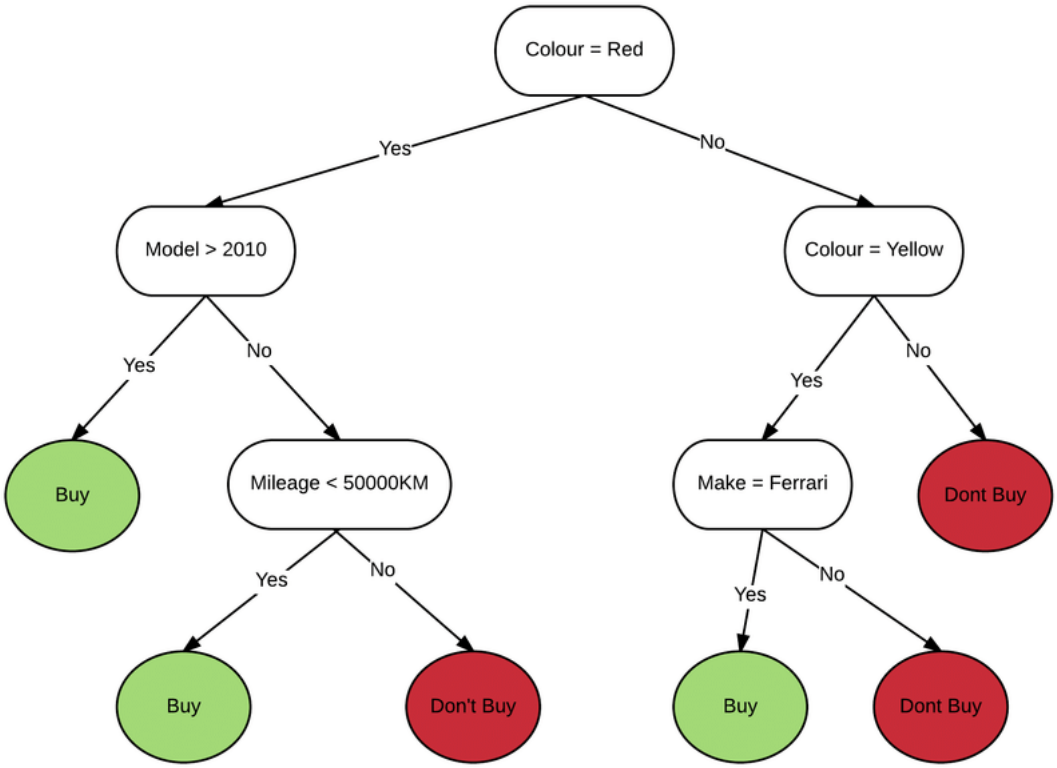


Fig. 2.18. Decision tree to decide whether or not to buy a car.

The random forest employed in this thesis is made up of decision trees that implement the "Gini impurity" in order to measure the quality of a split. Gini impurity is the probability of misclassifying a random sample from the data set if these were randomly labeled following the distribution of classes of the data set. Gini impurity, G, is calculated as follows:

$$G = \sum_{i=1}^{C} p(i) \cdot (1 - p(i)) \tag{2.20}$$

where C is the number of classes and $p(i)$ is the probability of randomly picking a sample from class i. Impurity 0 is the best and it is obtained when all samples of the subgroup

belong to the same class. Therefore, in order to measure the quality of a split, we want to maximize Gini's gain [28].

Random forest algorithm combines a large number of decision trees that are uncorrelated. Each of the decision trees will perform a different prediction and the prediction with the most votes is the prediction picked by the random forest. The fact that decision trees are uncorrelated is crucial since this allows the trees to perform predictions together that are more accurate than the individual prediction of each tree. With this approach, trees protect each other from their individual errors. There are two methods used by random forests to guarantee low or null correlation between trees:

- **Bagging:** Decision trees are very sensitive to the data that they are trained with. If the training data changes, the resulting tree can have a very different behavior. With the use of bagging, a random forest trains decision trees feeding each tree with randomly crafted subgroups of the training data. This results in very different decision trees and with very low correlation among them. This randomly crafted subgroups give a somewhat stochastic behavior to the classifier. The distribution employed to create the subgroups is uniform hence, each sample from the data set has the same probability to be picked [29].

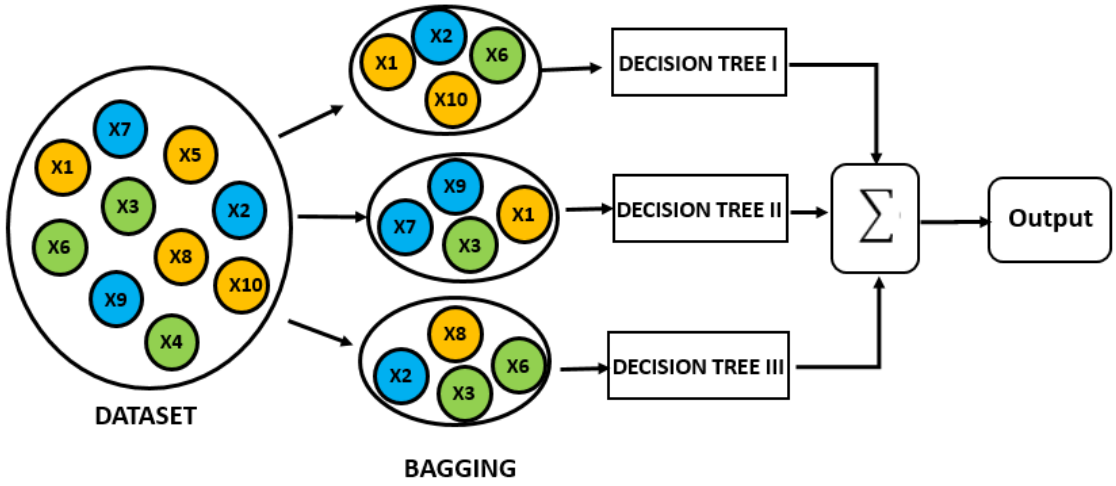


Fig. 2.19. Bagging example.

- **Randomly assign features:** In a normal predictor, all characteristics are evaluated to then pick the ones that have the best performance in splitting up classes into subgroups. However, if each tree in a random forest considers only a random subset of characteristics such as the one in figure 2.20 then the resulting trees have less correlation among theme [30].

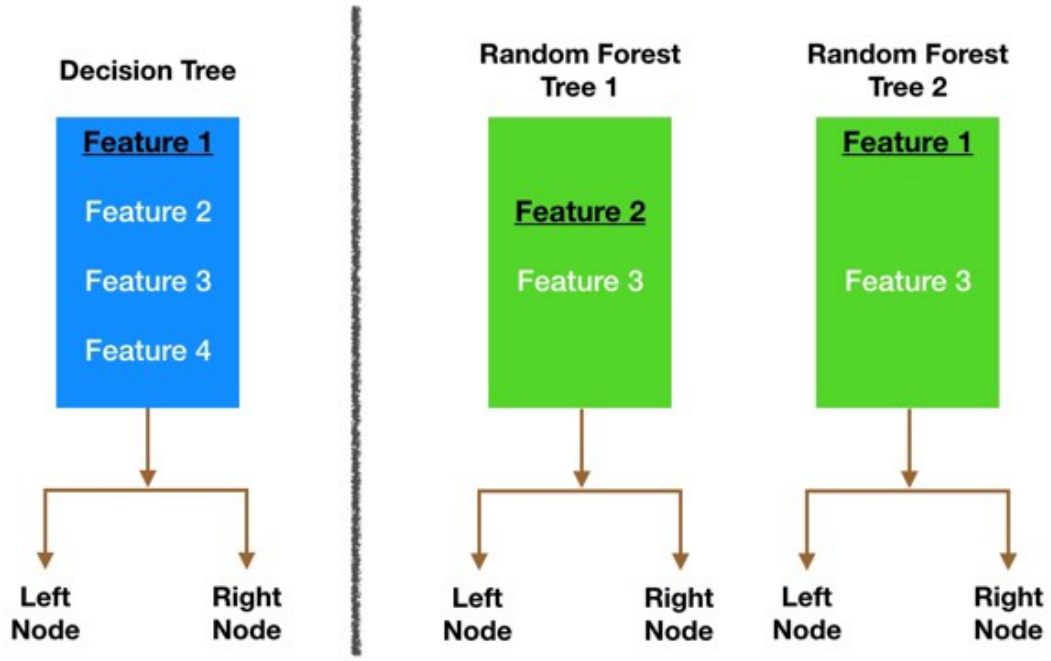


Fig. 2.20. Node splitting based on a random subset of features for each tree.

This thesis performs hyper-parameter tuning to minimize the error rate in the random forest classifier. The first parameter to optimize is the max depth of each tree, with the goal of limiting the training time, computational complexity and the overfitting [1]. The second parameter is the number of trees to employ in the forest, since it is inefficient to use more trees than necessary hence the lowest number of trees is searched for.

Lastly, it is worth mention the advantages of random forest to other classifiers. Since there are no suppositions, data does not have to be normalized nor preprocessed. Furthermore, no dimensionality reduction is need since trees end up using only the most relevant characteristics.

XGBoost

XGBoost, also known as eXtreme Gradient Boosting [31], is one of the most dominant algorithms in the field of machine learning. It shines due to its effectiveness and scalability in modern machine learning system which use huge amounts of data, such as the fraud-detection systems operated by banks. It operates ensemble learning since it combines multiple models with the objective of solving a problem with a better performance than the obtained by the any of combined models by theirselves. The algorithm is accesible

[1] In machine learning, an over fitted model is the result of over training an algorithm on a training set. The problem is that the over fitted model classifies perfectly the training set but it is not able to generalize hence the testing set perfomance is very poor

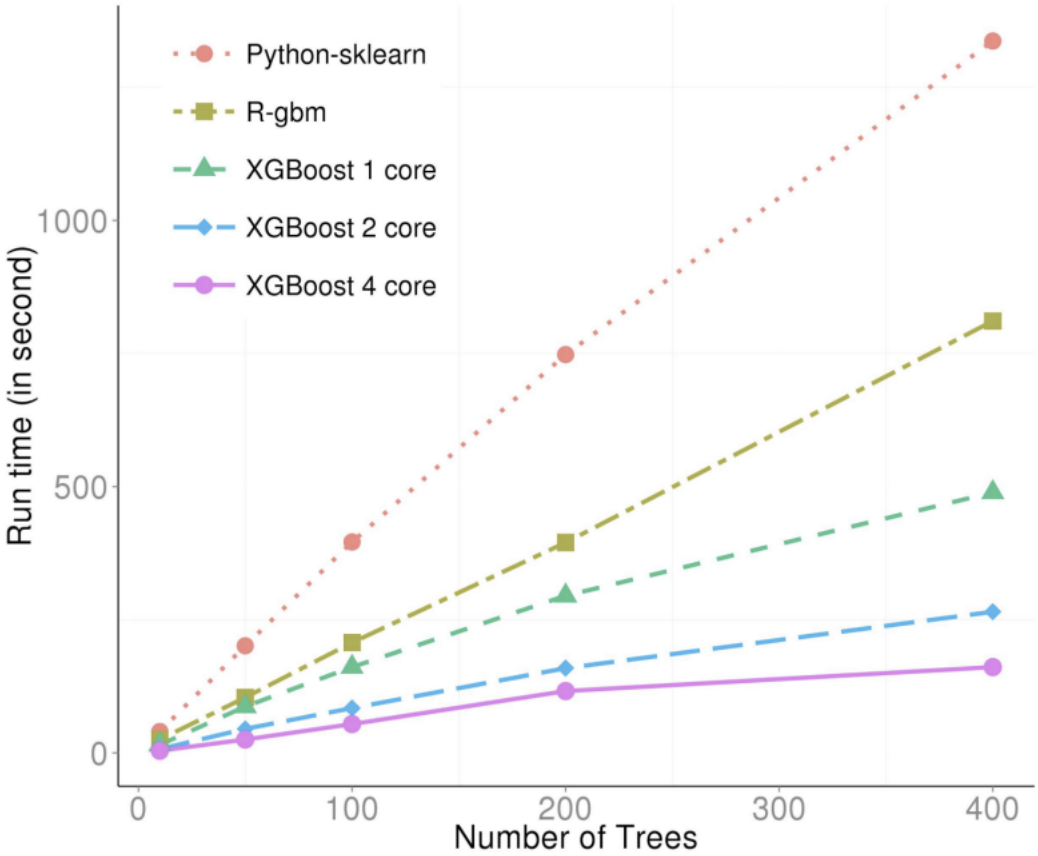


Fig. 2.21. Comparison between XGBoost and other classifiers

through the xgboost library, created by the researcher and professor at the University of Washington, Tianqui Chen.

Its principal advantages over other algorithms are the ability to parallelize the construction of trees using all the CPU cores and the ability to perform distributed training within a cluster. The figure 2.21 shows a comparison between the XGBoost algorithm, the random forest implementation from scikit-learn, and a generalized boosted regression model. The figure shows the number of trees trained as a function of time. Its visible that despite the computational cost of utilizing multiple cores in parallel, the model is able to spawn and train 400 decision trees in a few minutes.

The classifier implements the gradient boosting algorithm with decision trees. The idea behind this algorithm is to create a very powerful decision rule combining weaker rules. In order to achieve this behavior, new decision trees are add upon the existing ones to fix the errors that the existing ones are making until there is no room left for improvement. During this process, at time t, a lower weight is assigned to the outputs of the correctly classified samples at $t - 1$, and a higher weight is assigned to the outputs of the misclassified samples. In figure 2.22 a simple example is shown to demonstrate how boosting is employed in order to combine weak rules and generate a more reliable model. In addition, this gradient boosting algorithm also focuses on creating new models that amend the errors made by the previous ones. In order to achieve this, it uses the

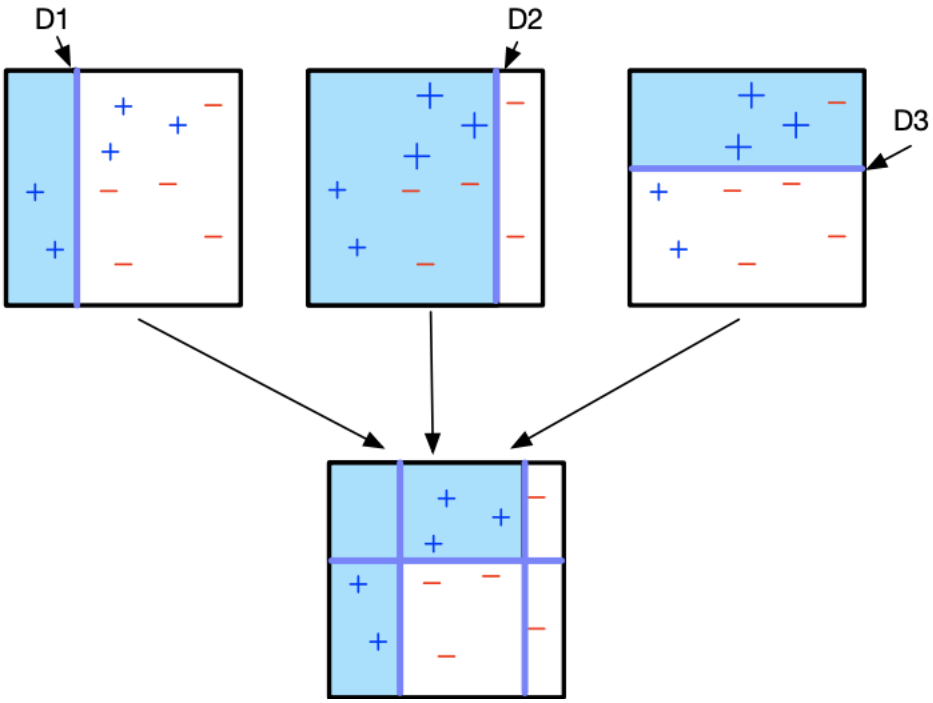


Fig. 2.22. Simple example of the boosting algorithm in a binary classification problem

gradient descent algorithm in order to minimize the error when new models are added. The advantage of this is that is useful for both classification and regression. Similarly to a random forest, each decision tree uses the bagging method to ensure a low correlation among them.

The main hyper-parameters to optimize are the maximum depth of the decision trees and the learning rate, which is used to lower the weights after each epoch in the boosting process. To conclude, the XGBoost classifier is the most computationally expensive classifier after deep multi layer perceptrons.

K-Nearest Neighbors

The K-Nearest Neighbors [32] is one of the simplest and most intuitive classifiers and its employed in a wide variety of fields, such as finance, image recognition and noise recognition. Each sample in the test set is classified to the majority class of the k nearest samples, hence, the training set is used straight away on the testing phase. In order to operate in this manner, the training set has to be saved in memory during the testing phase. This shows a clear advantage over other classifiers since it does not require to have a big training set to create a model. It works both for classification an regression problems as long as distances among samples can be computed.

More in detail, for a given training set with labeled samples $x = [x_0, x_1, ..., x_N], y$ the goal is to classify an unknown sample q. In order to do this, the k nearest samples to q from the training set have selected. This is done by measuring the distance between the **x** features of q and the **x** features of the rest of the samples. The subset of k closest samples its called training subset. It is necessary to normalize the features to a mean of 0 and

variance 1 in order to avoid large valued features have more weight than others.

The employed classifier in this thesis uses the Euclidean distance since the features are continuous:

$$d(x_i, q) = \sqrt{(x_{i,1} - q_1)^2 + (x_{i,2} - q_2)^2 + \ldots + (x_{i,L} - q_L)^2}, \quad i = 0, \ldots, N \tag{2.21}$$

where x_{ij} is the *j-th* feature of the *i-th* sample from the training set, L is the number of employed features, N is the total number of samples and q_i is the *i-th* feature of the sample q.

Once the k closest samples have been selected, the majority emotion among those k samples is chosen. The design only requires the optimization of the hyper-parameter k, that represents the number of near neighbors as it can be seen in figure 2.23 where by modifying the value of k we obtain a different classification.

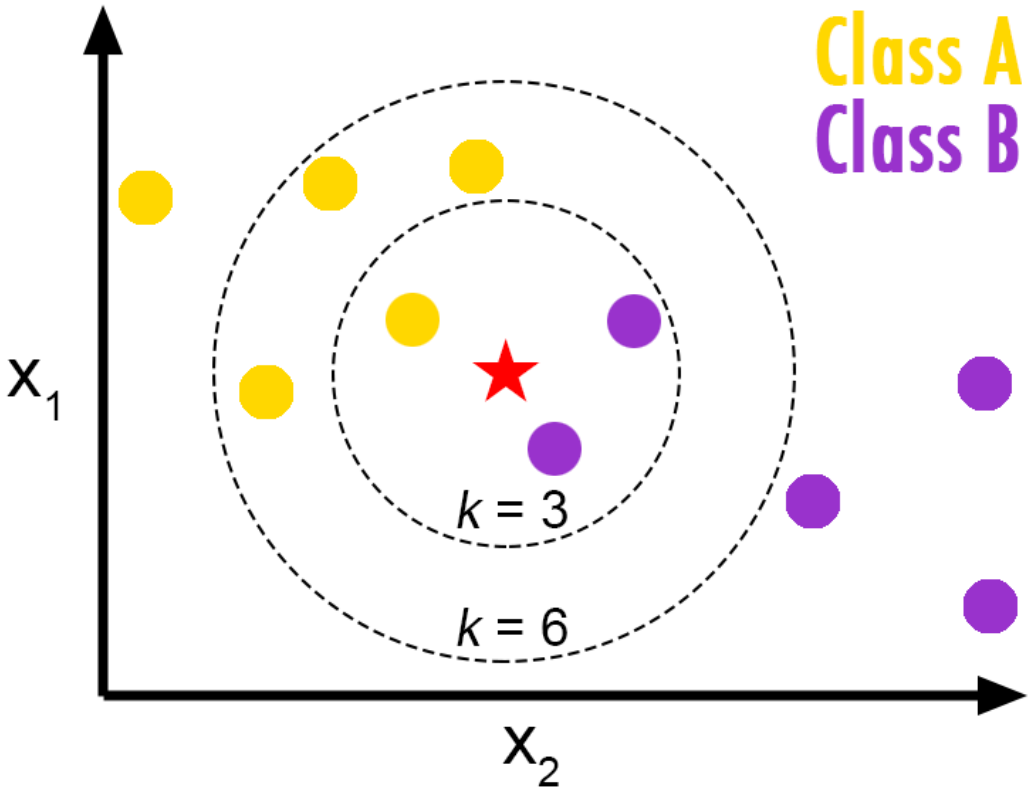


Fig. 2.23. Binary classification with k-NN, $k = 3$ and $k = 6$

Naive Bayes

Naive Bayes models are a special type of machine learning models. This algorithm is based on a technique for statistical classification known as "Bayes' Theorem". This models are called "naive" since they assume that the variables in play are independent from each other, meaning that the presence of a feature inside a data set is not related to the presence of any other characteristic. This algorithm allows to easily create models due to its simplicity.

The basis of the algorithm is to calculate the posterior probability of an event A happening given a set of probabilities from previous events.

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)} \tag{2.22}$$

where $P(A)$ is the probability of A, $P(B|A)$ is the probability of B knowing that A has previously occurred, $P(B)$ is the probability of B, $P(A|B)$ is the posterior probability of A happening given R.

Fig. 2.24. Naive Bayes unrolled

The algorithm for classification is a kind of supervised learning technique since it needs the labels to learn the probabilities. In order to apply the algorithm the following steps need to be taken:

1. Create a table of frequencies of each sample from the training set
2. Create a table of probabilities for each of the possible events as seen in figure 2.24.
3. Use the Naive Bayes equation (2.22) in order to calculate the posterior probability of each class
4. The class with the highest calculated probability is the result of the prediction.

The strengths of this algorithm are that is a fast a simple way of solving binary and multi-class classification problems, in the case features are independent from each other its performance is very good, high dimensional data does not affect the performance since assuming features are independent from each other results in the treatment of class distributions as if they were only dependent on one variable. On the other hand, the weaknesses of the Naive Bayes are that even-though this algorithm is a great classifier its estimating skills are very poor, it does not perform in most of real world data since the assumption of feature independence rarely holds in real life. In addition, when the test set has a sample

with a feature not seen during training the model will assign a probability of 0 making the model useless.

Support Vector Machines (SVM)

Support Vector Machine is an supervised learning algorithm that is employed in many real life classification and regression problems, such as biomedical applications, natural language processing and speech or voice recognition.

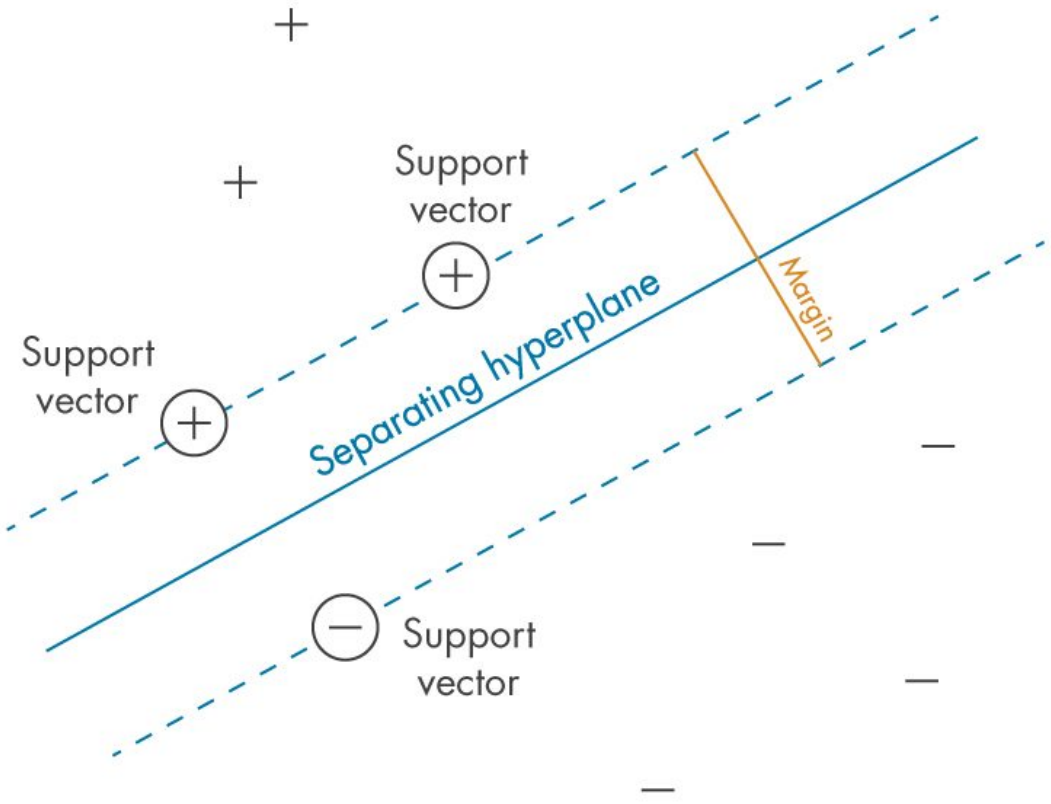


Fig. 2.25. Margin of a SVM [33]

The main goal of an SVM is to find the hyper-plane that best splits the samples belonging to two different classes. "The best split" implies that hyper-plane uses the widest margin between both classes, represented in figure 2.25 by plus and minus symbols. The margin is defined as parallel hyper-plane to the separating hyper-plane which does not contain any samples between itself and the separating hyper-plane. The algorithm can only find this separating hyper-plane in the problems that can be solved by linear separation; which in most practical problems does not happens, hence, the model maximizes the margin allowing for some small number of misclassifications ("soft margin") [33].

The support vectors refer to a subset of training samples that identify the location of the separating hyper-plane. The basic SVM algorithm is designed for binary classification problems. However, it can be easily adapted to multi-class problems by splitting the problem in a set of binary problems.

SVMs belong to a kind of machine learning algorithms known as kernel methods, in which a kernel function is used to transform characteristics. Kernel functions map data points to a different dimensional space, which is usually higher, with the hope of having more easily separable data inside that higher dimensional space. This simplifies the complex non-linear decision boundaries by making them linear in the higher dimensional space. The neat thing about the algorithm is that it does not require to actually transforms the data points to a higher dimensional space, which is tremendously computationally expensive, it avoids this by ussing the kernel trick[33].

When designing this algorithm we need to optimize a set of hyper-parameters to obtain the best performance in our specific task. First the kernel to use needs to be chosen; radial-base kernel, linear, polynomial, sigmoid. Then the penalty for each misclassified data point, C, needs to be selected which will affect the regularization of the model. Then in the case radial-base function, polynomial or sigmoid kernels are employed, the gamma value needs to be selected, which controls the distance of influence of a single data point in training [34].

3. SYSTEM FOR DIMENSIONAL AND DISCRETE EMOTION RECOGNITION FROM SPEECH

This chapter explains the implemented system for speech emotion recognition. Firstly, the selected data sets are detailed. Afterwards, the feature extraction and processing of both audio and text are reviewed aswell as the different extracted features such as MFCCs, fundamental frequencies, text embeddings and others. Lastly, the design and implementation of the thesis proposed systems is described. Both systems employ machine learning techniques to operate which range from Deep Learning architectures like LSTMs for regression to classifiers such as Random Forest, XGBoost, Support Vector Machines and others.

3.1. DATA ANALYSIS

3.1.1. Data sets

The systems implemented learn based on a given set of training data and are evaluated on a set of testing data as it is thoroughly explained in chapter 2. Hence, the data set the system operates on requires to have expressive audio files with labeled emotions assigned to each file. The performance of the models is highly positively correlated with the amount of samples since the models capacity to generalize [2] increases with the number of samples. Moreover, it is convenient that samples are balanced, meaning that there are no emotions or label values of VAD which count with a significant bigger amount of samples than the other. Another important prerequisite is that the system should be able to correctly behave with both male and female voices, therefore the data set used needs to have a good balance between male and female speakers. Furthermore, since the model is going to operate with both audio and text is also convenient that the data set comes with transcriptions for the audio files.

The employed datasets are: IEMOCAP: Interactive emotional dyadic motion capture database (English) [35] and MSP Podcast data set (English) [36]. This allows for evaluation of models with different languages to the learned one.

IEMOCAP: Interactive emotional dyadic motion capture database.

This data set is an english acted database collected by the SAIL lab at the University of Southern California. The database is formed by 16kHz samples which are in different formats such as audio, transcriptions, video and motion-capture recordings[35]. The database englobes a total of five sessions that last in 12 hours in which ten actors are

[2] It is said that a model generalizes correctly if its able to correctly predict the outputs of data not used during the training phase

involved.

The data set is split in two different data points, dialogues and utterances. Dialogues are acted scenes where actors perform improvisations of affective situations or follow a script to represent some emotions. On the other hand, utterances are smaller segments of the longer dialogues which correspond to each of the actor's interventions during a dialogue. In this thesis since the objective is to infer the emotion of only one individual, the utterances data points are used, in total 10,039.

Transcription	Start (s)	End (s)	Emotion	Valence	Arousal	Dominance
What's happening?	8.0737	11.47	neu	3.0	2.5	2.5

Table 3.1. IEMOCAP UTTERANCE EXAMPLE

Each of these utterances, table 3.1, are annotated in three different forms. The first form is emotional categorical evaluation which is performed by at least three annotators, in which categories range over the set of: angry, happy, sad, neutral, frustrated, excited, fearful, surprised, disgusted, other. In the case annotators could not come to a consesus on which emotion to annotate, the utterance is marked with "xxx". Secondly, dimensional emotional annotations are given by at least two annotators, the axes employed are: valence (positive vs. negative), arousal (calm vs. excited) and dominance (in control vs. submissive), evaluation is done by averaging the different evaluations for each of the axis of every annotator which range from 1 to 5 in all axis. Lastly, utterances have motion capture annotations which allow to perform analysis of the facial expressions, head and hand movements. However, since the thesis focuses on the audio signal, videos and motion capture annotations are left out.

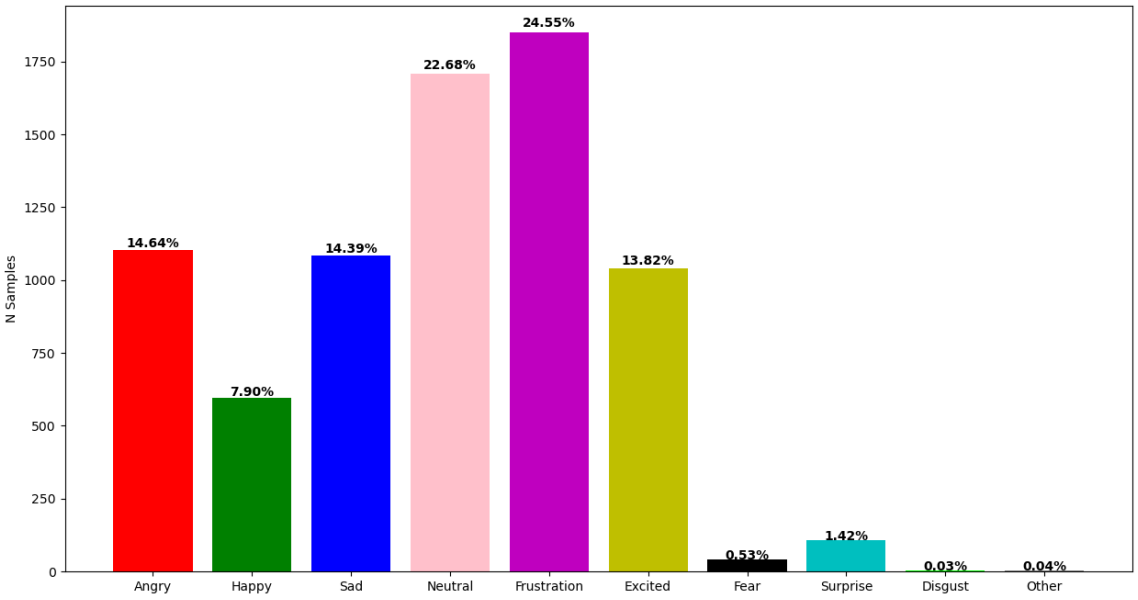
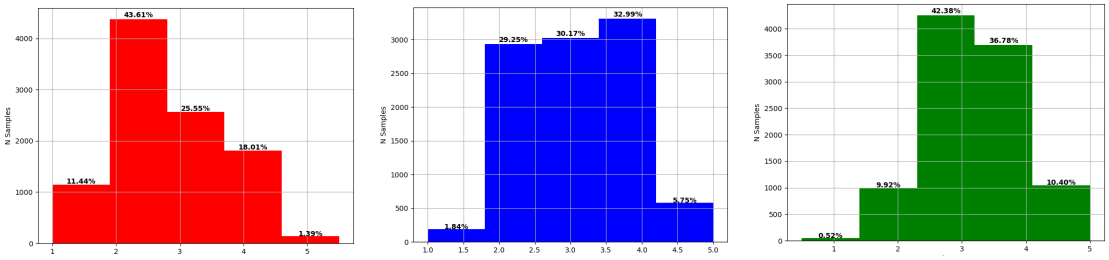


Fig. 3.1. Balance of discrete emotions for IEMOCAP

Ir can be seen in figure 3.1 that there is some problems with class imbalances which are addressed further in the thesis. In terms of dimensional emotions, it seems like they have a fair distribution according to the distribution emotion classes as it can be seen in figure 3.2.

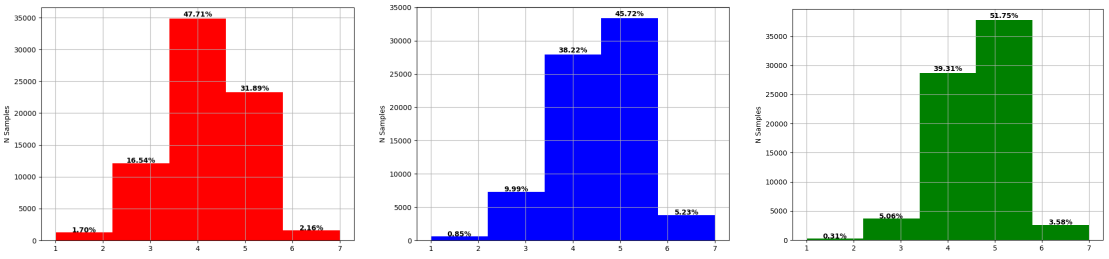


(a) Valence (b) Arousal (c) Dominance

Fig. 3.2. Balance of dimensional emotions for IEMOCAP

MSP Podcast

In contrast to the IEMOCAP data set, the MSP Podcast data set is formed by natural interactions between english speaking people instead of actors. To obtain these naturalistic samples the authors extracted 16kHz audio segments that contained emotional content from different multimedia providers over the Internet such as YouTube, Vimeo, Facebook, Instagram and more. The segments come from different conversations between people with various backgrounds over a large range of topics such as political debates, movie reviews, sports and more. To assure balanced emotional content, single speaker segments and noise free audios, the creators processed the samples with algorithms to detect voice, speaker diarisation and noise level estimation. In total there are 113 hours of audio which correspond to the 73,042 sentences/segments and 1,285 different speakers.



(a) Valence (b) Arousal (c) Dominance

Fig. 3.3. Balance of dimensional emotions for IEMOCAP

As it can be seen in figure 3.4 in this case classes are more balanced with only neutral and happy class considerably imbalanced, which can be easily fixed by undersampling. In addition, a distribution of the dimensional emotions can be seen at figure 3.3.

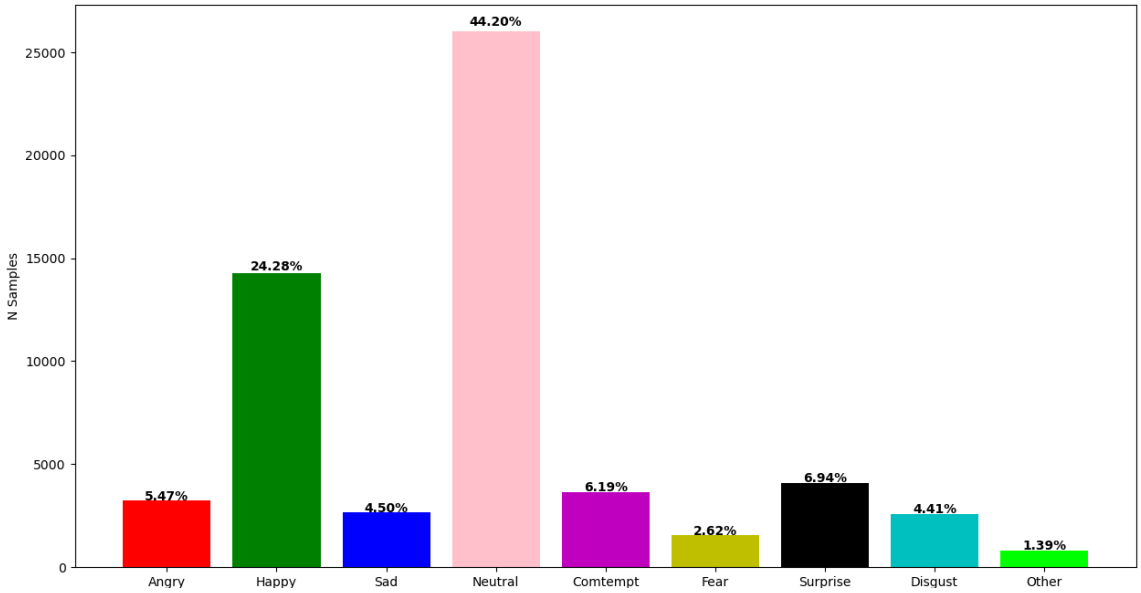


Fig. 3.4. Balance of emotions for MSP

Since samples are annotated via a crowd sourcing platform, the creators followed a method that tracks the performance of the annotators and when it falls below an acceptable threshold, they stop working. Annotations of segments include 2 types, dimensional emotions and categorical emotions. For the dimensional emotions a Self-Assessment Manikin [36] is followed where valence, arousal and dominance range from 1 to 7. In the case of categorical emotions these 9 are included: anger, sadness, happiness, surprise, fear, disgust, contempt, neutral, other and when there was no agreement among annotators "X" is written down. The data set also includes secondary emotions, however, this thesis uses only the primary emotions just described. An example of an annotated sample can be seen in table 3.2 where the transcription is obtained from the audio file with the use of the google audio to text API.

Transcription	Emotion	Valence	Arousal	Dominance
I feel validated by the soundtrack of the film	happy	5.8	2.4	3.6

Table 3.2. MSP UTTERANCE EXAMPLE

3.1.2. Audio features and processing

This section describes how audio is transformed into a set of features or characteristics that represent the information of the sound that will be the input to the system. All audio files have to be processed in this manner since they belong either to the training set or the testing set.

Signal theory defines a continuous signal *a(t)* as a signal with an amplitude *a* at a time *t*, *a* can be an audio signal that comes from the current generated by microphone

membranes vibrations. However, this continuous signal has to be discretised and quantised. This is due to the fact that the signal is going to be processed by a digital system and this type of systems can only store finite amount of values. Discretization is known as Nyquist-Shannon sampling in which a continuous time signal is sampled at a sampling frequency f_s in order to obtain a N fixed samples per unit of time from *a(t)*. The relationship between discrete time n and continuous time t is given by the sampling period T_s. Afterwards, the signal *a(n)* continuous amplitude values need to be mapped to a finite set of numbers yielding *x(n)*. This process is known as quantisation and introduces an error because the continuous values are mapped to the nearest discrete values in the set making the real values unrecoverable. The precision of the used system is *b=16* giving 2^b or 65,536 possible values [37].

$$T_s = \frac{1}{f_s} \tag{3.1}$$

$$t = n \cdot T_s. \tag{3.2}$$

Moreover, the discrete signal *x(n)* can be represented in the **frequency domain** by going through a Discrete Fourier Transformation (DFT). The signal after the DFT is *X(m)*, also known as the spectrum of *x(n)* where m is the discrete frequency $m = \frac{f}{f_0}$

$$X(m) = \sum_{n=0}^{N-1} x(n) e^{\frac{-j2\pi mn}{N}}. \tag{3.3}$$

. However, since the human ear is only subsceptible to the magnitudes of *X(m)* and not the phase, the analysis will only be done over the magnitude *Xm(m)*.

$$X_M(m) = |X(m)| = \sqrt{Im(X(m))^2 + Re(X(m))^2}. \tag{3.4}$$

$$X_\phi(m) = arctan\left(\frac{Im(X(m))}{Re(X(m))}\right). \tag{3.5}$$

Since DFT's computation has a complexity of O(N$\hat{2}$), in real systems, the Fast Fourier Transform is implemented which follows a divide and conquer approach. This approach achieves a complexity of O(Nlog(N)) but requires the frame size N to be a power of two, and when is not the case it transforms the frame by padding with zeros to the next higher power of two [37].

$$f_0 = \frac{1}{N \cdot T_s}. \tag{3.6}$$

Furthermore, audio signals contain very important information in both spectrum, *X(m)*, i.e, speaker's pitch, and time domain *x(n)*, i.e, amplitude and loudness. However of the most important issues in audio feature extraction is that all these attributes change over time hence performing an analysis over the whole signal is very computationally expensive. Therefore, **short-time analysis** is performed instead of a unique global analysis. This method, known as windowing, splits the audio signal into a set of N frames or short-term windows which are desired to be stationary segments of the original signal. The step

size is the time between the start and end of a frame. If the step size is lower than the frame size, the frames will be overlapping. Whereas, if step size and frame size are the same, there is not any overlapping. Typical values for the frame length range from 10 to 100 milliseconds. For example, if a step size of 10 milliseconds is used in combination with 40 millisecond frame length, it would result in a 75% overlap [38]. Basically, this framing methodology corresponds to the multiplication of the signal *x(n)* with a window function $x_r(n)$.

$$x_k(n) = x(k \cdot N_f^{(T)} + n) \cdot w_r(n). \tag{3.7}$$

There are different types of window functions:

- **Rectangular window**: Constant function for 0 to N-1 with amplitude equal to 1 for all values of n. In frequency domain corresponds to the sinc function

  $$W_{Rec}(m) = \frac{\sin(m)}{m} = sinc(m) \tag{3.8}$$

  Very convenient for time domain analysis when extracting Zero Crossing Rate and amplitude descriptors. In addition, is very efficient but not recommended to use in spectrum analysis [37].

- **Hanning window**: Known ass the raised cosine window. It is symmetric in the time domain and reaches zero amplitude at both sides. Defined from 0 to N-1 [37].

  $$w_{Han}(n) = 0.5\left(1 - \cos\left(\frac{2\pi n}{N-1}\right)\right) \tag{3.9}$$

- **Hamming window**: Variation of the Hanning window where side lobes amplitude is reduced and does not reach zero amplitude at the sides. It is the most common function on speech recognition and analysis. Defined from 0 to N-1 too.[37]

  $$w_{Ham}(n) = \alpha - \beta\cos\left(\frac{2\pi n}{N-1}\right). \tag{3.10}$$

Other very important concepts in signal theory are **amplitude** and **energy**. Signal amplitude ranges from maximum to minimum amplitude, and usually when amplitude is 0 means no signal is at the input of the system. However there can be a DC offset due to characteristics of equipment used to record the audio signal. In some cases this DC offset can provide intersenting information since ir corresponds to the mean of *x(n)*. If *x(n)* is assumed to have no DC offset, the signal energy of *x(n)* is:

$$E = \sum_{n=0}^{N-1} x^2(n). \tag{3.11}$$

. However, in the speech processing field is more common to employ the Root Mean Squared energy and the logarithmic energy:

$$E_{rms} = \sqrt{\frac{1}{N}\sum_{n=0}^{N-1} x^2(n)}. \tag{3.12}$$

$$E_{log} = E_{bias} + E_0 \cdot \log \sum_{n=0}^{N-1} x^2(n), \tag{3.13}$$

MFCCs

MFCCs or Mel-Frequency Cepstral Coefficients are a type of cepstral coefficients employed by musicians and producers to model audio and music. It has been the most widely used feature for speech recognition due to its ability to represent the voice amplitude spectrum in a compacted form. These coefficients are the result of a applying a Cepstrum over a window frame of an audio signal. This Cepstrum is an operator that transforms a time convolution into a sum in the frequency domain, extracting both excitation and vocal tract out of a signal. It is defined as the inverse Fourier transform of the signal spectrum in logarithmic scale.

$$\text{Cepstrum}\ (s[n]) = \hat{s}[n] = F^{-1}[\log(|F[s[n]]|)] = \frac{1}{2\pi}\int_{-\pi}^{\pi} \log\left(\left|S\left(e^{jw}\right)\right|\right) dw \tag{3.14}$$

where $s[n]$ represents the convolution between excitation and vocal tract.

$$s[n] = e[n] * h[n] \tag{3.15}$$

Nevertheless, this operator is not used straight away. In figure 3.5 there is a basic diagram showing the basic process for the MFCCs extraction. After the DFT of the convolution between the excitation and the vocal tract, the information is mapped to the Mel scale using a filter bank. The purpose of this change in scale is to try to imitate the human's hearing psycho-acoustic behavior. Mel scale is characterized for having better resolutions for lower frequencies alike humans.

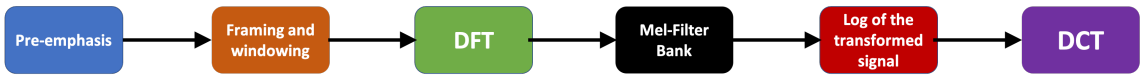


Fig. 3.5. MFCCs extraction process

In a study carried out by the researcher Beth Logan, at the Cambridge Research Laboratory [39], 3 hours of a news program where recorded in which 10% of audio was music and 90% was voice. MFCCs were extracted with frames of 25.6 milliseconds and an overlapping of 10 milliseconds. Logan's study then compared if it was more appropriate to map the information into the Mel scale or, in contrast, it was better to stick to the linear model for a classification problem between voiced and musical segments. Splitting the data into 2 hours for training and 40 minutes for testing and extracting 13 coefficients per frame resulted in better performance than the linear model. However, it is uncertain whether the Mel scale models better voiced segments or using another frequency range would be beneficial. What can be assured is that Mel scale does not have a negative impact on human voice classification problems.

Now each of the blocks of figure 3.5 are described:

1. **Pre-emphasis**: The audio signal undergoes a pre-emphasis filter to compensate the attenuation of -20db/decade that results from the human voice production model. It is important to apply this type of filter since it allows to emphasize intensity peaks in the spectrum (formants) generated at high frequencies where most of the acoustic energy is concentrated.

2. **Windowing**: Typically, Hamming and Hanning window functions are used. In this thesis a Hanning window with a frame size of 20ms and overlapping of 10ms is used. Therefore, MFCC coefficients are obtained every 10ms.

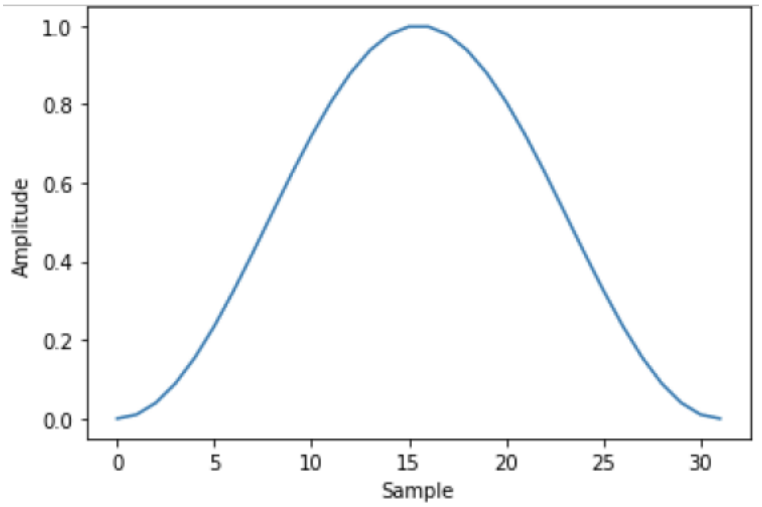


Fig. 3.6. Hanning Window

3. **Discrete Fourier Transform**: After applying the window function, the DFT is operated over the signal and from that moment on only the modulus of $X(m)$ is used.

4. **Mel filter bank**: The signal is multiplied by bank of triangular filters of unit area, spaced according to Mel frequencies.

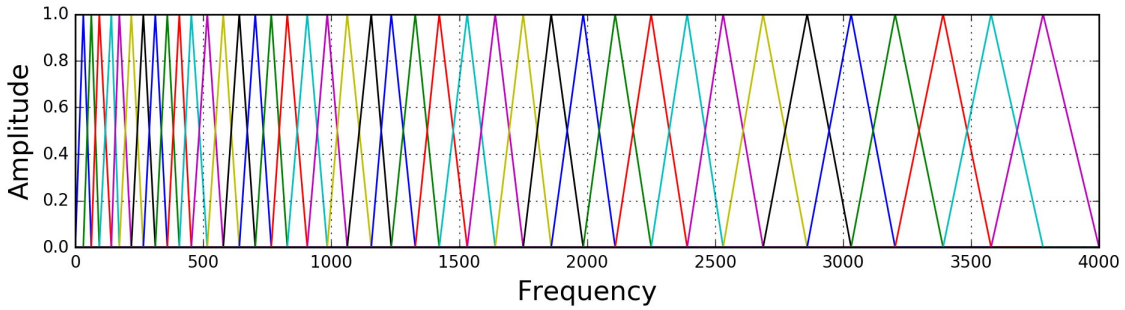


Fig. 3.7. Mel filter bank

$$\text{mel}(f) = 2.595 \log_{10}\left(1 + \frac{f}{700}\right) \tag{3.16}$$

Although the standard is to use 26 filters, in this thesis a filter bank containing 40 filters has been utilized since is the most common amount of filters in speech processing for a voice signal with frequency 16kHz [40]. Nevertheless, the amount of employed filters should not influence the final result since what matters is that the output of the Fourier transform has enough points to distribute the number of filters among them. A window of 20ms results in 320 samples, 160 real, which are enough to distribute among the 40 filters.

This scale represents a mapping between the frequency and the perceived tone. As it can be seen the human does not perceive tone in a lineal manner. From 0 to 1kHz the mapping is approximately linear and logarithmic for larger frequencies than 1kHz.

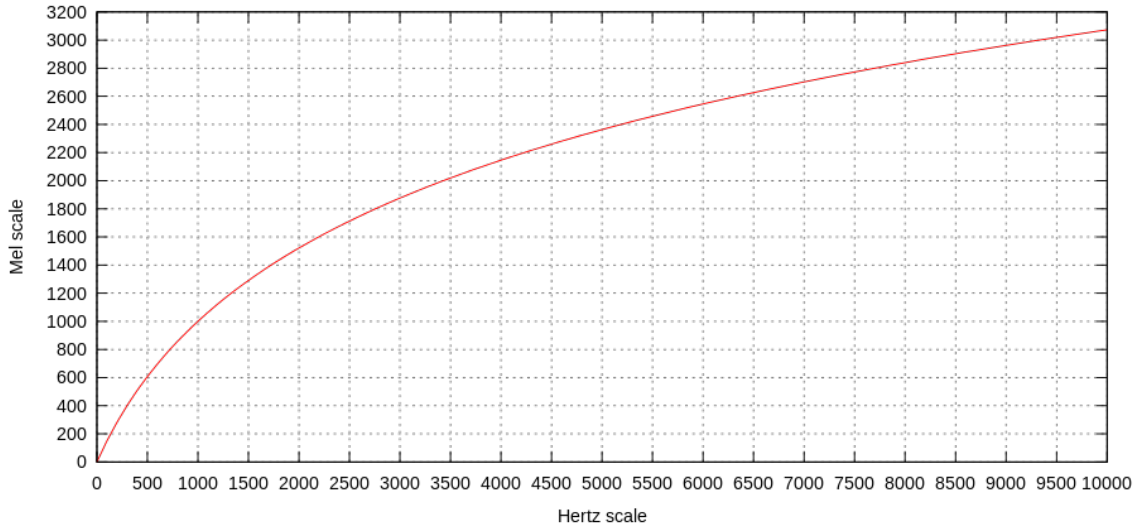


Fig. 3.8. Mel scale

It is also interesting to obtain the mathematical expression that allows to calculate the value for each filter:

$$H_m[k] = \begin{cases} 0, & k < f[m-1] \\ \frac{2(k-f[m-1])}{(f[m+1]-f(m-1])(f[m]-f[m-1])}, & f[m-1] \leq k \leq f[m] \\ \frac{2(f[m-1]-k)}{(f[m+1]-f[m-1])(f[m]-f[m-1])}, & f[m] < k \leq f[m+1] \\ 0, & k > f[m+1] \end{cases} \tag{3.17}$$

where $1 < m < 40$, m is the filter of each filter where:

$$f[m] = \left(\frac{N}{F_s}\right) B^{-1}\left(B(f_1) + m\frac{B(f_h) - B(f_1)}{M+1}\right) \tag{3.18}$$

$$B^{-1}(b) = 700\left(e^{\frac{b}{2595}-1}\right) \tag{3.19}$$

F_s is the sampling frequency (16 kHz), f_1 is the lower end of each triangular filter and f_h the top end [41]. After multiplying the modulus of *X(m)* with the mel filter bank, the energy corresponding to each of the filters is calculated:

$$E_m = \sum_{k=0}^{N-1} |x[k]|^2 H_m[k] \quad 1 \leq m \leq 40 \tag{3.20}$$

5. **Logarithm of transformed signal**: Once the energy has been calculated, the logarithm it taken to transform it into the logarithmic power spectrum. This step is necessary since the human ear does not perceive sounds in a linear scale. Then average cepstral subtraction is performed which is a normalization technique that allows the extracted characteristics are more loyal to what humans really hear.

6. **Discrete Cosine Transform (DCT)**: Since the filters in our filter bank are overlapping, the energy from the adjacent filters are highly correlated. In order to avoid statistical dependencies among filters, a DCT is performed over the logarithmic energy. DCT de-correlates the energies by transforming the spectral coefficients into the cepstral space and resulting into the MFCCs. Despite the fact that this operation outputs ass manu coefficients as filters were used, in this thesis only the first 13 of them are employed.

$$c_{\text{mfcc}}[m] = \sum_{k=0}^{N-1} \log(E_k) \cos\left(m\left(k - \frac{1}{2}\right)\frac{\pi}{N}\right) \quad m = 1, \ldots, 13 \tag{3.21}$$

Fundamental frequency (Pitch)

Fundamental frequency [42] is another interesting feature employed in this thesis due to the fact that it is one of the most important sound characteristics and a variable with a lot of emotional information. This frequency is the lowest one in the spectrum and the dominant frequencies can be expressed as multiples of the pitch.

The pitch is extracted by performing the same windowing method applied for the MFCCs, this time using a Hanning window, and followed by a quadratic interpolation of the signal's magnitude. This is a very powerful tool to estimate the instantaneous frequency close to a peak in the spectrum.

The magnitude peak is k_m. In the range k_{m-1} and k_{m+1}, the cuadratic interpolation employs a Lagrange polynomial to approximate the points $(km-1, \alpha)$, (km, β) and $(km+1, \gamma)$. The parabola has the following shape:

$$X(k) = a(k - \widehat{K})^2 + \widehat{X} \tag{3.22}$$

considering the vortex of the parabola $(\widehat{K}, \widehat{X})$ as an estimation of the sinusoidal parameters. The variable k is the location index and a is the only coefficient of the parabola. The expressions of $\widehat{K}$ and $\widehat{M}$ in terms of *km*,*alpha*,*beta* and *gamma* are:

$$\widehat{K} = k_m + \frac{1}{2}\frac{\alpha - \gamma}{\alpha - 2\beta + \gamma} \tag{3.23}$$

$$\widehat{X} = \beta + \frac{1}{8}\frac{(\alpha - \gamma)^2}{\alpha - 2\beta + \gamma} \tag{3.24}$$

The fundamental frequency is located at the maximum value of the approximate parabola. In figure 3.9, the parabolic approximation can be seen. The blue line represents the DFT of a Hanning window and the green line shows the approximated parabola [43].

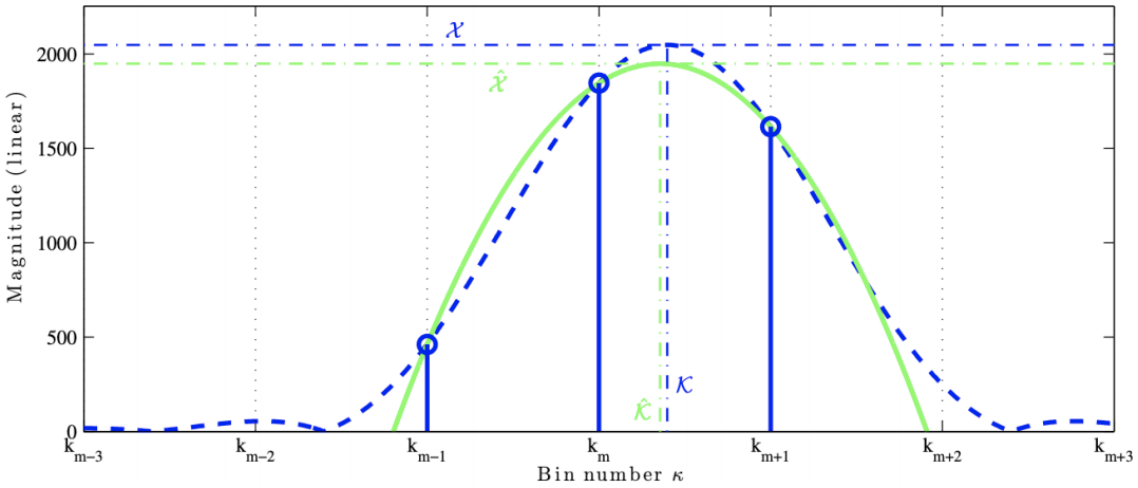


Fig. 3.9. Parabolic Approximation

Formants

Phonemes and vowels are characterized by resonance frequencies of the vocal tract system. These frequencies are kown as formants and they are seen as the maxima of the envelope. To extract these features a peak picking algorithm is employed over the speech power spectra. Nomenclature of formants uses the symbol F_i with $i > 1$ since they are, basically, higher orders of resonance frequencies excited by the fundamental frequency, F_0. When vowel sounds occur, they have most of its energy in the first three or four formants.[37]

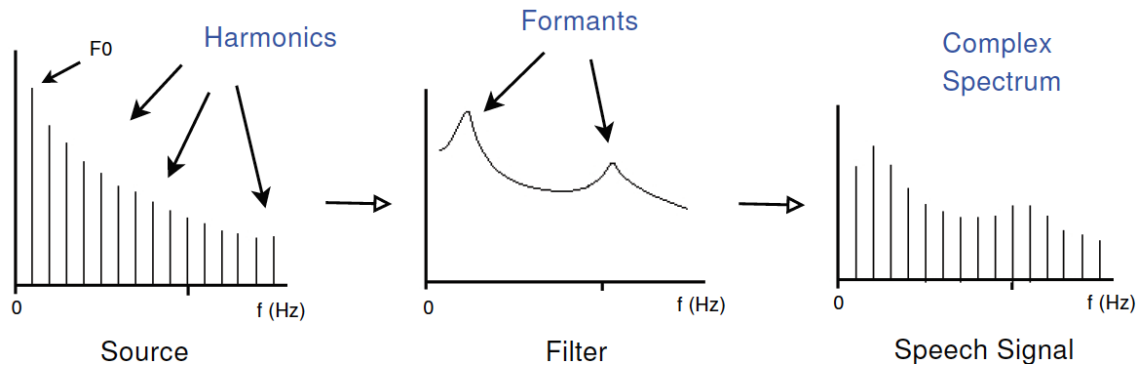


Fig. 3.10. Illustration of formants

Gaussian Triad

These features allow to calculate musical attributes from the audio signal such as dissonance, tension and modality of the chord. In addition, they can be used to compute: strength of weakest (π_{min}) and median (π_{med}) tones with respect to the tonic mode, the concentration of the constituents of chord with respect the tonic mode (s_{min}, s_{med}) and the intervals in terms of ratios (FR_1, FR_2). All of them from the representation of trimodal structure of ($F0$). Despite providing very good information about the audio signal, these features have a major downside which is that all temporal information is lost since variation of pitch along time is not taken into account [1].

In order to extract these features, an expectation-maximization algorithm has to be run over the interval distribution of the signal to obtain the minimum Gaussian mixture models that best approximate the distribution of the pitches. This process selects all dominant pitches and discards small pitch values as if they were noise, which in theory, retrieves the principal musical components of the signal.

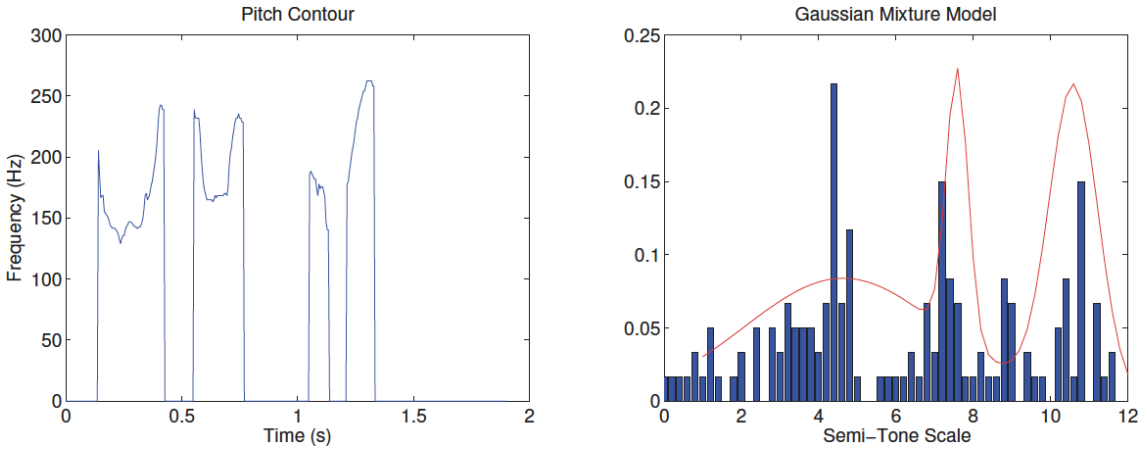


Fig. 3.11. Extraction of dominant pitches for a happy utterance

The number of optimal "clusters" can be obtained by using the Akaike information criterion, AIC. [44]. This criterion takes into account simplicity of the model, model's accuracy and includes a penalization term proportional to model's complexity to measure the fit of an statistical model. Therefore, the lower the AIC, the better the model.

$$\mathrm{AIC} = 2k - 2\ln(L) \tag{3.25}$$

where k is the number of model parameters and L is maximized value of the likelihood function [1].

Once the Gaussian mixtures are obtained, parameters can be defined. *Tonic mode* or Gaussian with highest amplitude, a_{ton} which is at postion k_{ton} and has standard deviation σ_{ton}. Median and minimum mixtures have a_{med}, a_{min} at k_{med}, k_{min} with standard deviatons

σ_{med},σ_{min}.

$$\begin{aligned}
\pi &= a_{ton} \\
\pi_{\min} &= \frac{a_{\min}}{a_{ton}} \\
\pi_{med} &= \frac{a_{med}}{a_{ton}} \\
s_{\min} &= \frac{\log \sigma_{\min}}{\log \sigma_{ton}} \\
s_{med} &= \frac{\log \sigma_{med}}{\log \sigma_{ton}} \\
FR_1 &= \frac{k_{med}}{k_{ton}} \\
FR_2 &= \frac{k_{\min}}{k_{med}}
\end{aligned} \tag{3.26}$$

Employed features

To conclude, the utilized audio features in this thesis are detailed in table number 3.3:

Table 3.3. EMPLOYED AUDIO FEATURES

Name	**Description**	**#**
Gaussian Triad	3.1.2	7
ZCR	Describes the number of sign changes c of $x(n)$ per unit of time	1
Energy	Sum of squares of signal values, normalized by the respecetive frame length	1
Energy Entropy	The entropy of sub-frames' normalized energies. It can be interpreted as a measure of abrupt changes	1
Spectral Centroid	Center of gravity of the spectrum	1
Spectral Spread	The second central moment of the spectrum	1
Spectral Entropy	Entropy of the normalized spectral energies for a set of sub-frames	1
Spectral Flux	Squared difference between the normalized magnitudes of the spectra of two successive frames	1
Spectral Rolloff	Defines as the frequency below n% of total energy is concentrated. It is computed from $X(m)$. In this thesis, $n = 25, 50, 75, 90$	4
MFCCs	3.1.2	14
Chromas	12 element representation of teh spectral energy where the bins represent the 12 equal-tempered pitch classes of western-type music	12
Chroma Deviation	Standard deviation of the 12 chroma coefficients	1
Loudness	Sum of the simplified auditory spectrum over all bands $X_{p,aud}(b)$ [37]	1
Shimmer	Amplitude variations of consecutive voice signal periods [37]	1

Harmonics-to-Noise Ratio (HNR)	Ratio between RMSEnergy of harmonic signal components over energy of the noise like components. $$HNR_{wf} = \frac{E_{\text{harm}}}{E_{\text{noise}}}$$	1
Relative energy of F1/F2/F3	Ratio of the energy of the spectral harmonic peak ath the 1st, 2nd and 3rd formant's centre frequency to the energy of the spectral peak at $F0$ [45]	3
Alpha Ratio	Ratio of the summed energy from 50-1000Hz and 1-5kHz	1
Hammarberg Index	ratio of the strongest energy peak in 0–2 kHz to the strongest peak in 2–5 kHz	1
Slope 0–500/500–1500 Hz	linear regression slope of the logarithmic power spectrum within 0-500Hz and 500-1500Hz	1
RelH1-H3	Ratio of energy between first and second harmonics, H1 and H2	1
RelH1-A3	Ratio of energy between the first harmonic, H1, and the highest harmonic in the third formant range, A3	1
Pitch ($F0$)	3.1.2	1
Frequencies F1/F2/F3	Center frequencies for the first, second and third fromants	3
Bandwidth F1/F2/F3	Bandwidths of the first, second and third formants	3
Voice prob	Raw voicing probability obtained from the SHS algorithm [37]	1
Jitter	Variation of the length of the fundamental period from one single period to the next [37]	1
RMSEnergy	3.1.2	1
Band Energy	Computed using a rectangular filter. Filter starts at f_l, lower frequency bound of the band, and ends at f_u, upper frequency bound of band. All magnitudes inside the filter are summed. Employed bands are 250-650Hz and 1-4kHz. [37]	2
RASTA auditory bands	Band raw values from the RASTA filtered auditory bands $X_{p,aud}(b)$	26
Spectral Moments	Spectral variance, skewness and kurtosis [37]	3
Spectral Slope	$X(m)$ can be approximated by a line. $$\widehat{y} = ax + b$$ where a is the spectral slope and can be computed with a minimum quadratic error approximation	1

psySharpness	Auditory sharpness, psycho-acoustically scales spectral centroid [37]	1
Spectral Harmonicity	Denotes the amount and quality of a signal's harmonics. It is computed by selecting the local minima and maxima in a frame, which are then used to compute their ratio in relation to the maxima's amplitude [37]	1
	Total Number of Audio Features	100

In order to efficiently extract the presented features from the audio database, software libraries were made use of. Specifically, this thesis mainly employs the programming language Python and its 3rd party libraries, *pyAudioAnalysis* and *openSmile*. In addition, the *Praat* scripting language is also used. The use of these software makes possible the automation of the audio feature extraction for all the retrieved samples.

	LLDs	**HSFs**
pyAudioAnalysis LLD: 34 HSF:68	Zero crossing rate, energy, entropy of energy, spectral centroid, spectral spread, spectral entropy, spectral flux, spectral roll-off 90%, 13 MFCCs, 12 chroma vectors, chroma deviation.	mean std
eGeMAPS LLD: 25 HSF:50	Intensity, alpha ratio, Hammarberg index, spectral slope 0–500Hz, spectral slope 500–1500Hz, spectral flux, 4 MFCCs, F0, jitter, shimmer, harmonics-to-noise ratio (HNR), harmonic difference H1-H2, harmonic difference H1-A3, F1, F1 bandwidth, F1 amplitude, F2, F2 amplitude, F3, and F3 amplitude.	mean std
ComParE 2016 LLD: 65 HSF:130	F0, voicing probability, jitter, shimmer, HNR, RMSEnergy, band energy 250-650Hz, band energy 1-4kHz, lengthL1norm, Zero Crossing Rate, 26 RASTA auditory bands, 4 spectral roll-off (25, 50, 75, 90)%, spectral flux, spectral centroid, spectral entropy, spectral variance, spectral skewness, spectral kurtosis, spectral slope, psySharpness, spectral Harmonicity, 14 MFCCs	mean std

Table 3.4. AUDIO FEATURE SETS

pyAudioAnalysis and *openSmile* are open-source libraries that provide multiple anal-

ysis procedures. However, in this thesis only the feature extraction functionality is be used. They implement high level interface to perform short-term analysis on wav audio files with window sizes from 20 to 100 ms [46]. This short term analysis produces what is known as Low Level Descriptors (LLDs). LLDs provide characteristics at frame level, which are usually 40ms long, and since they are very short period of times the resulting features for a whole audio are overwhelming and sometimes even confusing for the system. Therefore, it is typical to use High Statistical Functions (HSFs) which are the mean and standard deviations of these LLDs. Using HSFs is common practice and usually keeps audio's informative characteristics without producing an overwhelming number of data points. Both tools come with a preconfigured feature sets to extract which are inspired in studied and validated sets of features proposed by the literature. In this thesis, *pyAudioAnalysis* extracts the pyAudioAnalysis feature set and *openSmile* extracts the eGeMAPS (extended Geneva Minimalistic Feature Set) [45] and the ComParE 2016 (Computational Paralinguistics Challenge 2016) [47]. A summary of these acoustic feature sets can be seen at table 3.4.

3.1.3. Text features and processing

This section describes how text is transformed into a set of features or characteristics that represent the information of the sequences of words that will be the input to the system. All text have to be processed in this manner since they belong either to the training set or the testing set.

The first step is to obtain all the text related to each of the audio utterances, known as transcriptions. The IEMOCAP data set comes with transcriptions for each of the audios, so that is done. However, MSP-Podcast contains only audio files with no associated transcription. To generate the text transcriptions an audio to text algorithm is used. A transcription is generated for all the audio files using python, the library SpeechRecognition and the Google Cloud Speech API.

Machine learning models can not deal with textual representations or strings since no mathematical operations can be computed over them. Therefore, text has to be vectorized in some manner in order to allow models to gain information from text. However, this vectorization needs to be useful and vectors accurately represent reality, i.e, capture the semantic content of text. Text preprocessing is key to achieve this useful representation, it consists in previously cleaning, structuring and homogenizing text. The first step of preprocessing is tokenization which consists in splitting text in smaller pieces called tokens. In this thesis the biggest structures are utterances' transcriptions and tokens are words, punctuation, numbers, etc... After tokenization, many tokens have upper case and lower case letter, the same token sometimes appears in its singular form and sometimes in its plural form or the same verb token appears in different tenses. In order to semantically analyze text, tokens which are formally different but actually have the same meaning need to be homogenized. The most common homogenization steps are:

1. **Removal of upper case letters and non alphanumeric characters**: This approach all upper case letters are transformed to lower case and no alphanumeric characters are removed, i.e, interrogation marks.

2. **Stemming**: In everyday language, words take different shapes in terms of gender, quantity, time (in the case of verbs) and more. However, normalizing this various shapes into a canonical form is very handy for may applications. Stemming is a process that helps with this normalization by hard trimming a token to its stem, i.e, running after stemming is run.

3. **Lemmatization**: Its main objective is shared with stemming, normalization. However, instead of hard trimming a token to a stem, lemmatization uses lexical knowledge to obtain the correct basic forms of words or lemmas, i.e, women's lemma is woman.

The last preprocessing step is cleaning, which consists in removing irrelevant words or stop words from transcriptions. Stop words are the most common words in a language, i.e, in English: *the, to, at, over, ...* These words do not have a relevant meaning and are usually removed.

The following step after preprocessing is vectorization, which should create a numerical representation from the corpus (set of transcriptions) that preserves the semantic relationships among words. One of the most common representation is Bag of Words (BoW), which consists in the analysis of the frequency of words inside a transcription. It is called *bag* of words since any information about the order of structure of words inside text is thrown away. BoW does not care about where words appear inside a transcription, only cares about the frequency of the word. Creating a vocabulary, set of unique words, is the first step to create a BoW. Then vectorization is performed by creating a vector of size equal the vocabulary size for each audio transcription. At each position of the vector stores the number of times the word appears in the transcription. For example, for transcription 1 if the word "hello" is the 53th word in the vocabulary and appears 7 times in the transcription, then, the vector for transcription 1 has 7 at position 53. There is a big issue with the BoW representation, the frequency of very common words dominates the representation compared to the rest of the words. This makes the BoW value more common words such as commonly used verbs which may not provide as much as information to the model as strange words that are a lot more specific to the field of study. To solve this problem, TF-IDF is introduced. TF-IDF's proposal is to readjust the frequency of words taking into account their frequency among the whole corpus. This readjustment penalizes words that are common among the whole corpus and not only common inside a transcription. A high TF-IDF score of a word means that the word has a high frequency inside the transcription but a low frequency in the corpus.

$$\mathrm{TF}(w,t) = \frac{\text{number of times that } w \text{ appears in transcription } t}{\text{total number of words in transcription } t} \tag{3.27}$$

$$\text{IDF}(w, C) = \log \frac{\text{number of words in the corpus}}{1 + \text{ transcriptions where the word } w \text{ appears}} \tag{3.28}$$

$$\text{TF} - \text{IDF}(w, t, C) = \text{TF}(w, t) * \text{IDF}(w, C) \tag{3.29}$$

Although BoW and TF-IDF are simple and flexible vectorial representations to manage textual information, some limitations need to be taken into account:

- **Vocabulary**: Requires to be carefully design, specially in terms of its size which affects the scattering of transcriptions representations.
- **Scattering**: Sparse representations are harder to model both due to computational issues and information issues. When high scattering occurs the models have a very hard time to seize the little meaningful information that is lost inside such an enormous representation space.
- **Meaning**: Since the order of words is thrown away, context and semantics are ignored. This is problem since context and words' meaning can provide very meaningful information to the model. For example, it would be able to distinguish between *this is interesting* and *is this interesting?*.

These issues may lead to overfitting of the model since modelling a high dimensional space implies the use of many parameters. In addition, sparse representations limit the identification of similar concepts, for example, Rome and Paris would be at the same distance as Rome and Italy in such representations.

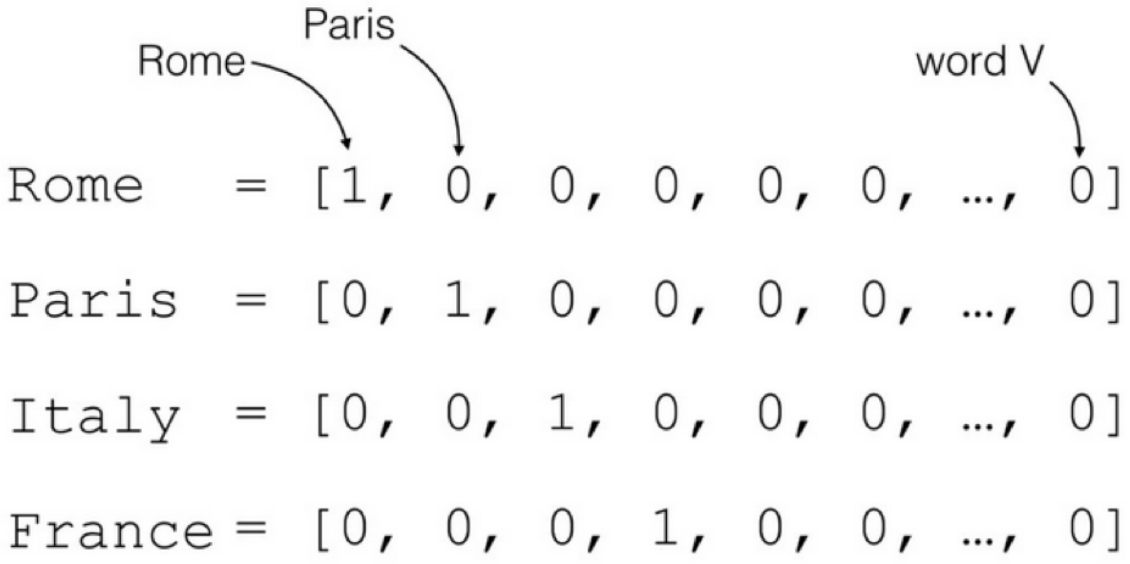


Fig. 3.12. One Hot Encoding Example

There exists a text representation technique that solves these issues called embeddings. Embeddings are a vectorial representation of data in a relatively low dimensional space. They are able to transform big one hot encode sparse vectors into a lower dimension vector that preserves semantical relationships among words since similiar cities, movies or other concepts have similar distances in the vectors space. Ideally, a good embedding provides a set of vectors whose postion (distance and direction) in the vectors space encodes the semantics of the words the vectors represent. Semantic relationships such as genre, verbal tense and geography can be seen in figure 3.13.

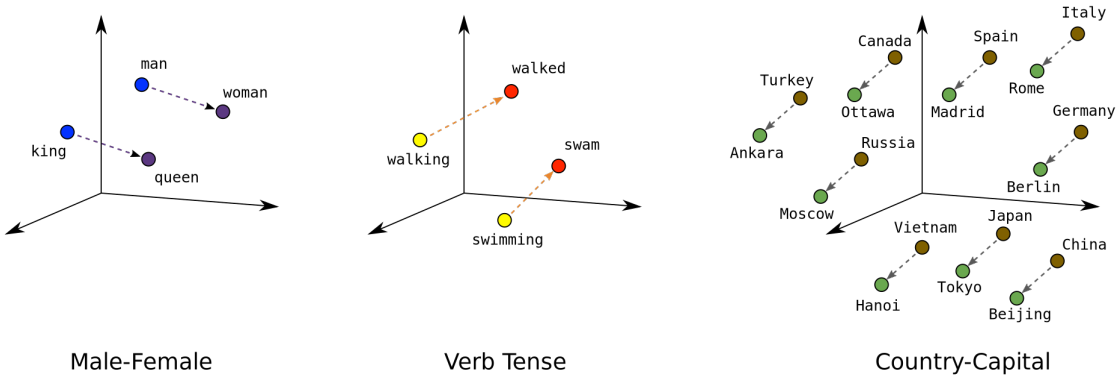


Fig. 3.13. Embeddings keeping semantic relationships

Therefore, representing data through embeddings has many advantages. Model's learning benefits from the lower dimensions of an embedding since they reduce the computational cost for learning and lower the chances of overfitting. Moreover, if the embeddings are well designed they provide more meaningful information to the model which is able to learn the underlying relationships of data easier. In addition, embeddings' lower dimension compared to frequency representations allows to have a better visual analysis of the vectors.

Furthermore, one of the best features from using embeddings is that they are reusable among applications. For example, GloVe embeddings are a set of pre-trained embeddings which can be reused. GloVe stands for Global Vectors, and obtains 300 dimensions vector representations of words through an unsupervised algorithm. *"The algorithm maps words into a meaningful vector space where distance between vectors is related to the semantic similarity of the words represented by the vectors. Training is performed on aggregated global word-word co-ocurrence statistics from a corpus"* [48]. The training to obtain these quality embeddings is extremely computationally expensive and time consuming. In order to avoid having to spend the time and money to obtain quality embeddings, is very common to use pre-trained embeddings and adapt them to the specific problem that is being faced. GloVe embeddings are used in this thesis for the english language, they are developed by Standford University [49].

3.1.4. Summary of extracted features

To conclude, after the feature extraction process the shape of the data is shown in table 3.5. As it can be seen there is a lot more data from the MSP-PODCAST data set. However, this amount of data is too much for the computational capabilities of this thesis. Furthermore, this data set is much more dirty since its constructed with recordings from podcasts whereas the IEMOCAP data set includes recordings from actors that try to explicitly elicit a set of emotions. Therefore, the thesis conducts the main experiments on the IEMOCAP data set and leaves the MSP data set for transfer learning purposes.

Data set	Audio	Text	Vocabulary
IEMOCAP 10,039 audios	**pyAudioAnalysis** 34 LLDs 68 HSFs **eGeMAPS** 25 LLDs 50 HSFs **ComParE 2016** 65 LLDs 130 HSFs	**Longest Sequence** 554 word **Embedding** GloVe 300-d vectors	**Embedding Coverage** 99.52% **Size** 3,438 word **Language** English
MSP-PODCAST 73,042 audios	**pyAudioAnalysis** 34 LLDs 68 HSFs **eGeMAPS** 25 LLDs 50 HSFs **ComParE 2016** 65 LLDs 130 HSFs	**Longest Sequence** 360 words **Embedding** GloVe 300-d vectors	**Embedding Coverage** 94.18% **Size** 31,273 word **Language** English

Table 3.5. FEATURE SETS AND MAIN CHARACTERISTICS OF IEMOCAP AND MSP-PODCAST DATA SET

3.2. IMPLEMENTATION AND DESIGN OF SYSTEMS AND MODELS

This section explains the workings and inner structure of the two built systems as well as the architectures and configurations of the models used in such systems. Both systems share the same goal, which is correctly classify a discrete emotion and predict its 3 dimensional representation in the valence-arousal-dominance space from an audio file.

3.2.1. Sequential System

This system in figure 3.14 is composed of two main steps. First a regression model is fed with both audio and text features from each audio file and performs regression analysis on the 3 output variables, VAD. Afterwards, a classifier model is fed with the output VAD from the previous regression and classifies each VAD into a discrete emotion.

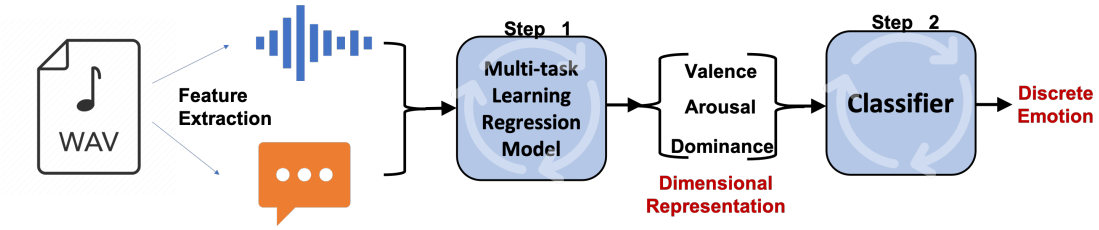


Fig. 3.14. Sequential system for dimensional and discrete emotion recognition

Multi-task Regression: Architecture (1st Step)

As discovered by the authors of [50], a bi-modal network fusion architecture gives the best performance when performing regression on the three output variables valence, arousal and dominance. The designed and implemented architecture in this thesis is formed by 3 main smaller networks, an acoustic network, a linguistic network and a concatenation network. The output of the acoustic network is concatenated with the output of the linguistic network. This approach is based on the assumption that human perception processes acoustic and linguistic information differently, it is also beneficial for the model that with the chosen architecture linguistic and acoustic feature sets do not need to have the same dimensions.

- **Acoustic Network:** The acoustic features that are extracted from audios are processed by this network which outputs a higher dimensional representation of the input. The first step of the network is to apply batch normalization on the input. Once the batch input is normalized, it enters a LSTM stack of 3 layers from which the full sequence output from the last layer is kept. Afterwards the output from the LSTM stack is flattened with a dropout probability *p_acoustic*.

- **Linguistic Network:** This networks processes the text features extracted from the audio transcriptions and outputs a higher dimensional representation of the input. The first step of the network is to pass the input through an embedding layer. The embedding layer performs a mapping from integer tokens generated from text into 300 dimensional vectors, i.e, pre-trained GloVe embeddings. This first step reshapes the dimensions of the input from (batch_size, sequence_length, num_words_per_seq) into (batch_size, sequence_length, num_words_per_seq * embedding_dim) where $num\_words\_per\_seq = 1$ most of the times. After this mapping into a better semantic representation of the input, the vectors enter a LSTM

stack of 3 layers from which only the output from the last timestep in the LSTM is kept. Next, the LSTM stack output enters linear layer which reduces dimensionality and whose output is dropped with a probability *p_text*.

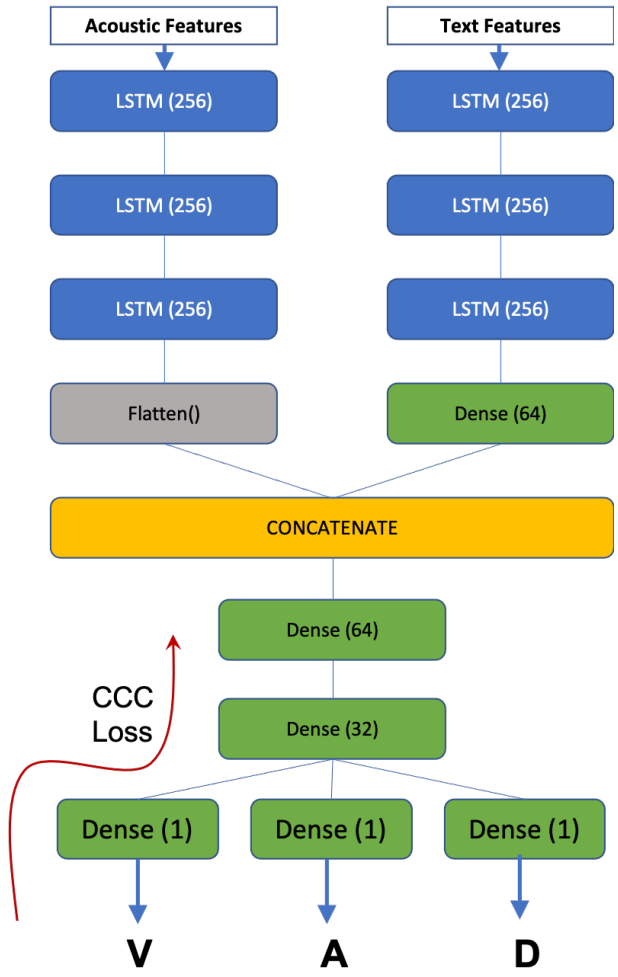


Fig. 3.15. Architecture of a MTL regression model using LSTMs for both the acoustic and text networks. V: Valence, A: Arousal, D: Dominance

- **Dense Network:** It concatenates the outputs from the acoustic network and the linguistic network to perform operations over them. The first layer in the network concatenates the the inputs with a dropout probability of *p_concat*. The concatenated input is passed through linear layer to reduce dimensionality. The function that this first linear layer implements is seen in equation 3.30.

$$f(y) = W_2 g\left([W_{1a}^{\top} xa + b_{1a}; W_{1t}^{\top} xt + b_{1t}]\right) + b_2 \tag{3.30}$$

where $f(y)$ denotes the output of the layer; weights from the previous acoustic (a) and text (t) layers are denoted as W_{1a} and W_{1t}; weigths from the current linear layer is W_2; x_a and x_b are acoustic and text features respectively; b is a bias and g is the Rectifier Linear Unit activation function. In the next step another linear layer is applied to reduce dimensionality more and implements a similar function

to equation 3.30 but now previous weights are from the previous linear layer and input is not feature and acoustic concatenated but the output of the previous layer. Lastly, the output from the second layer is splitted into 3 different linear layers. Each one of these last layers applies the same function as the previous layers and reduces dimensionality down to a point in space to perform regression individually on the three different output vairables: valence, arousal and dominance.

Multi-task Regression: Configuration (1st Step)

The configuration of this model encompasses the definition of a loss function for network learning and the scaling of the input data to the model. The loss functions this model optimizes is based on the CCC score which was introduced by Lin's calculation [51]

$$CCC = \frac{2\rho_{xy}\sigma_x\sigma_y}{\sigma_x^2 + \sigma_y^2 + \left(\mu_x - \mu_y\right)^2} \tag{3.31}$$

where ρ_{xy} is the Pearson correlation coefficient between x and y, σ is the standard deviation and μ is the mean. CCC ranges from -1, perfect disagreement, to 1, perfect agreement. Therefore, the loss function is defined to maximize the agreement between true and predicted emotions.

$$CCCL = 1 - CCC$$

The authors of [50] for this regression problem demonstrated that multi-task learning (MTL) is superior to single task or variable learning (STL). MTL is a type of learning that tries jointly learn multidimensional targets. For example, if the problem targets are y_1, y_2 then MTL optimizes both variables instead of optimizing only y_1 or y_2 which is known as STL. In the case of STL the loss function would optimize for either one of valence, $CCCL_V$; arousal, $CCCL_A$ or dominance; $CCCL_D$. In this thesis MTL is used for making the model simultaneously learn the three emotional dimensions with the use of total CCC loss, CCC_{tot}, that combines all CCC loss functions for VAD values.

$$CCCL_{tot} = CCCL_V + CCCL_A + CCCL_D.$$

However, as mentioned by the authors of [50], this is function does not accurately represents the variables relationships in reality. Therefore, parameters are introduced to weight arousal and valence, where the weight for the dominance is obtain by subtracting arousal and valence weights to 1.

$$CCCL_{tot} = \alpha CCCLV + \beta CCCLA + (1\text{-}\alpha\text{-}\beta)CCCL_D$$

To optimize this loss function RMSProp is employed. The traditional stochastic gradient (SGD) descent has a problem in that learning rates have to scale with 1/T to get convergence, where T is the iteration number (ie: after a while the model takes really small steps and does not get a lot of progress). RMSProp and gets around this by automatically adjusting the step size so that the step is on the same scale as the gradients.

Therefore, as the average gradient gets smaller, the coefficient in the SGD update gets bigger to compensate.

The model performs MaxMin scaling to the output targets to convert them from the different scales 1-5 (IEMOCAP) and 1-7 (MSP) to the range [-1, 1]. The conversion si done following these equations:

$$\begin{aligned} x_{std} &= \frac{x - x_{\min}}{x_{\max} - x_{\min}}, \\ x_{\text{scaled}} &= x_{std} \times (\max - \min) + \min \end{aligned} \tag{3.32}$$

where x is the original target score, *max* and *min* are both 1 and -1. This helps the model generalize better since it suppresses the effects of outliers in the data.

Multi-class Classifier: Algorithms (2nd Step)

This model's purpose is to classify the 3 output variables from the regression, valence-arousal-dominance, into the 6 discrete emotional space. Section 3.2.1 explains the reasons that lead these number of emotions.

In order to achieve this mapping different machine learning algorithms which are able to classify are employed to then compare their performance. These are common algorithms that this thesis employs: Random Forest, Multi-Layer Perceptron Network, XGBoost, Support Vector Machines, Naive Bayes Classifier and K-Nearest Neighbors. In 2.4 a detailed description for each of the algorithms can be found.

Essentially what this algorithms will be doing is approximating the functions that define the surfaces' shapes which include each of the emotions. These shapes can be very complex and hence the functions that define them can be very difficult to approximate. In principle, following the theory from the psychological model of emotions [13], the centers for the surfaces of each emotion should be positioned as presented in table 3.6. Once the shapes are approximated by a classifier algorithm, the model can classify a 3 dimensional point, VAD, into on of the 6 basic discrete emotions based on the surface that surrounds such point.

	Valence	Arousal	Dominance
Neutral	0.0	0.0	0.0
Anger	-0.43	0.67	0.34
Joy	0.76	0.48	0.35
Surprise	0.4	0.67	-0.13
Disgust	-0.6	0.35	0.11
Fear	-0.64	0.6	-0.43
Sadness	-0.63	0.27	-0.33

Table 3.6. DISCRETE MODEL TO DIMENSIONAL MODEL RELATIONSHIPS

Multi-class Classifier: Configuration (2nd Step)

The configuration of the classifiers encompasses the manipulation of the data accordingly to the stated problem, the definition of a loss function for the learning of the multilayer perceptron, the scaling of the input data and the definition of an evaluation metric (accuracy).

Due to the heavy imbalance in the employed IEMOCAP data set, the thesis focuses on classifying the 6 basic emotions (Happiness, Disgust, Anger, Fear, Surprise, Sadness) as it is proposed by the authors of the data set [35]. Therefore, the emotions 'Frustration' and 'Other' are removed. In addition, instead of removing excitement from the data, the thesis follows the advice from the data set authors and combines 'Excitement' with 'Happiness' in order to obtain more balanced data, since they are close in the valence and activation domain [35]. Finally, since 'Disgust' represents only 0.03% of the data set it has only been left out of the classification problem. These leaves the models with only 6 emotions to predict: anger, happiness, sadness, neutral, fear and surprise as it can be seen in figures 3.16 and 3.17.

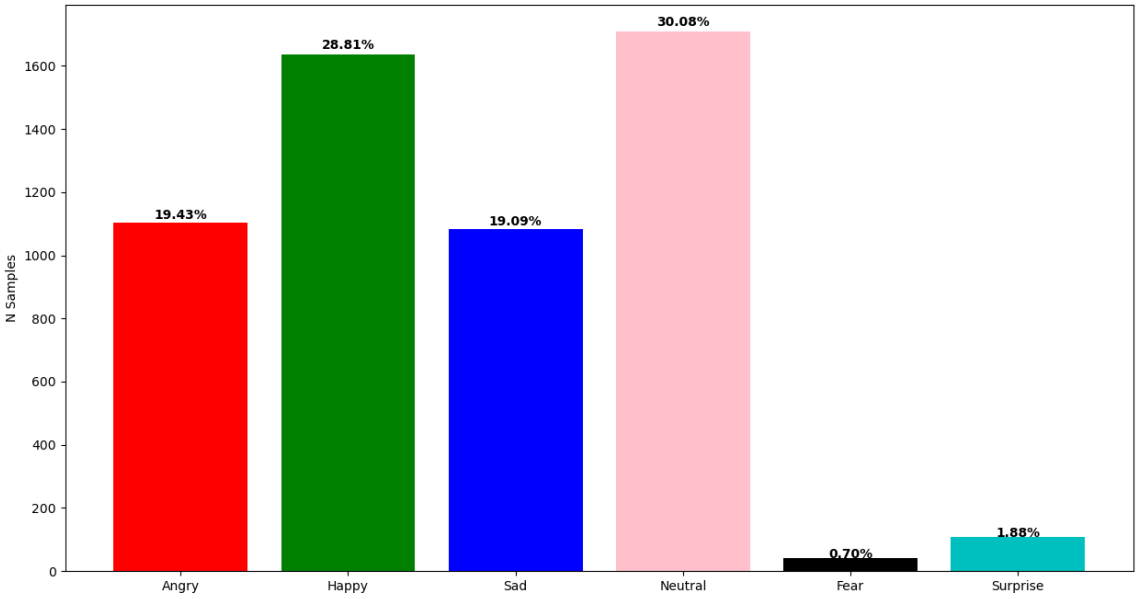
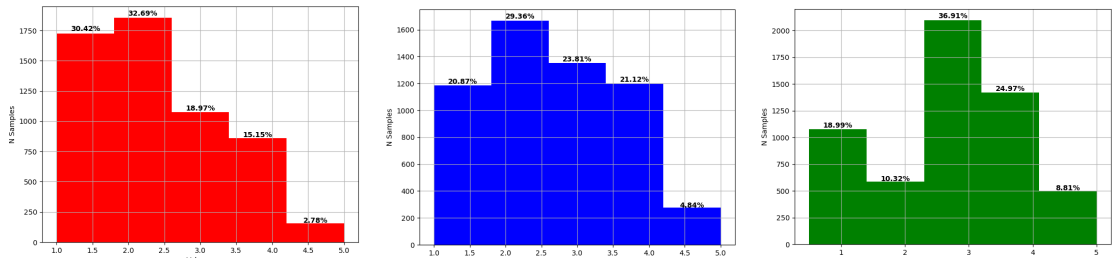


Fig. 3.16. Balance of discrete emotions for IEMOCAP after cleaning the data set



(a) Valence (b) Arousal (c) Dominance

Fig. 3.17. Balance of dimensional emotions for IEMOCAP after cleaning the data set

In the case of MLP, each index of the output vector corresponds to probability of a sample belonging to each discrete emotion. The problem needed to solve is a multi-class classification problem, hence, the typical loss functions that operate on binary classes are not handy. The employed loss is known as Cross-Entropy Loss or Log Loss which compares each predicted class probability with the actual class desired output 0 or 1. The CE Loss penalizes in a logarithmic fashion, yielding a large score for large differences close to 1 (bad) and small for small differences that tend to 0 (good). The perfect model has a CE Loss of 0.

$$\text{loss}(x,\ \text{class}\ ) = -\log\left(\frac{\exp(x[\ \text{class}\ ])}{\sum_j \exp(x[j])}\right) = -x[\ \text{class}\ ] + \log\left(\sum_j \exp(x[j])\right) \tag{3.33}$$

where x is the output vector with probabilities for each class and *class* is the index of class. However since the data sets acquired for this thesis are not perfect, the samples are not balanced. Imbalanced samples mean that some classes have a bigger representation, more samples labeled as one class, than others. This affects the model learning since it assumes a balanced training data set and leads to sub optimal classification where the model has a good performance on majority classes but minority ones are misclassified frequently. To solve this problem, the loss has to be adapated to a weighted CE Loss in which each class is assigned a weight to manipulate how importance should be given to a class when learning.

$$\text{loss}_{\text{weighted}}(x, \text{class}) = \ \text{weight}\ [\text{class}]\left(-x[\ \text{class}\ ] + \log\left(\sum_j \exp(x[j])\right)\right) \tag{3.34}$$

$$\text{weight[class]} = \frac{\max(\#\ \text{occurrences in most common class})}{\#\ \text{occurrences of class}}$$

The rest of algorithms have already been discussed in 2.4. For all algorithms the input vectors with ranges 1-5 (IEMOCAP) are MaxMin scaled in the range -1 to 1 for simplicity of the pipeline of all models even-though Random Forests are not sensitive to the variance in data.

In order to evaluate and compare all the proposed algorithms the thesis needs some metrics measure how well the classification task is performed. However, before introducing these metrics, some concepts need to be described. To begin with, a *positive class* can be "Angry" and the *negative class* in that case is "Not Angry". Furthermore, the following definitions are important: *True Positive (TP)* refers to a correctly classified sample of the positive class, *True Negative (TN)* refers to a correctly classified sample of the negative class, *False Positive (FP)* refers to a misclassified sample of the positive class and a *False Negative (FN)* refers to a misclassified sample of the negative class.

- **Accuracy:** Measures the percentage of the total samples that the model has correctly classified as seen in below's equation.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.35}$$

- **Precision:** Measures the percentage of positive classified samples that the model classified as true positive as seen in below's equation.

$$precision = \frac{TP}{TP + FP} \tag{3.36}$$

- **Recall:** Measures the amount of true positives over the number of total positive sampless as seen in below's equation.

$$recall = \frac{TP}{TP + FN} \tag{3.37}$$

- **F1 score:** Combines recall and precision scores into one metric, it is calculate using the harmonic of precision and recall as seen in below's equation.

$$F1score = 2 * \frac{precision * recall}{precision + recall} \tag{3.38}$$

All these metrics have a weighted version of themselves which takes into account class imbalances. The chosen metric to select the best model in classification task in this thesis is accuracy.

3.2.2. Parallel System

This system in contrast to the sequential one, classifies and performs regression in only one step as seen in figure 3.18. With this approach, both VAD values and the discrete emotion are predicted in parallel, i.e, at the same time and by the same model from the text and audio features input. The architecture of the system's model is essentially the same as the MTL regression model described in 3.2.1.

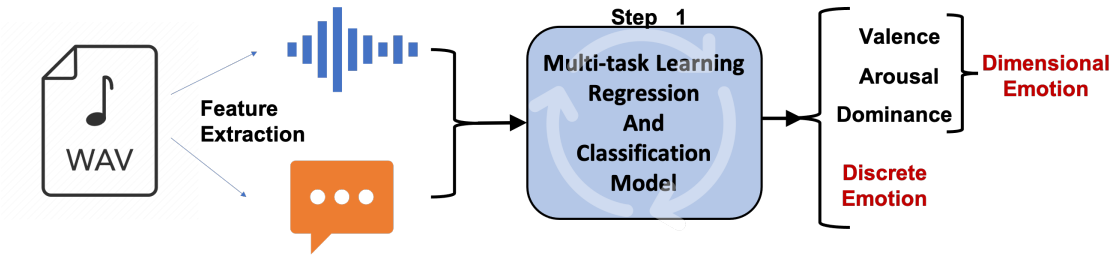


Fig. 3.18. Parallel system for dimensional and discrete emotion recognition

This system adds to the base architecture from 3.2.1 a new branch of dense layers, a second dense network, that comes out from the concatenation layer and whose output is a one-hot vector. Each index i of this output vector has a value that, when "softmaxed", represents the sample's probability of belonging to the *i-th* emotion. This architecture is shown in figure 3.19

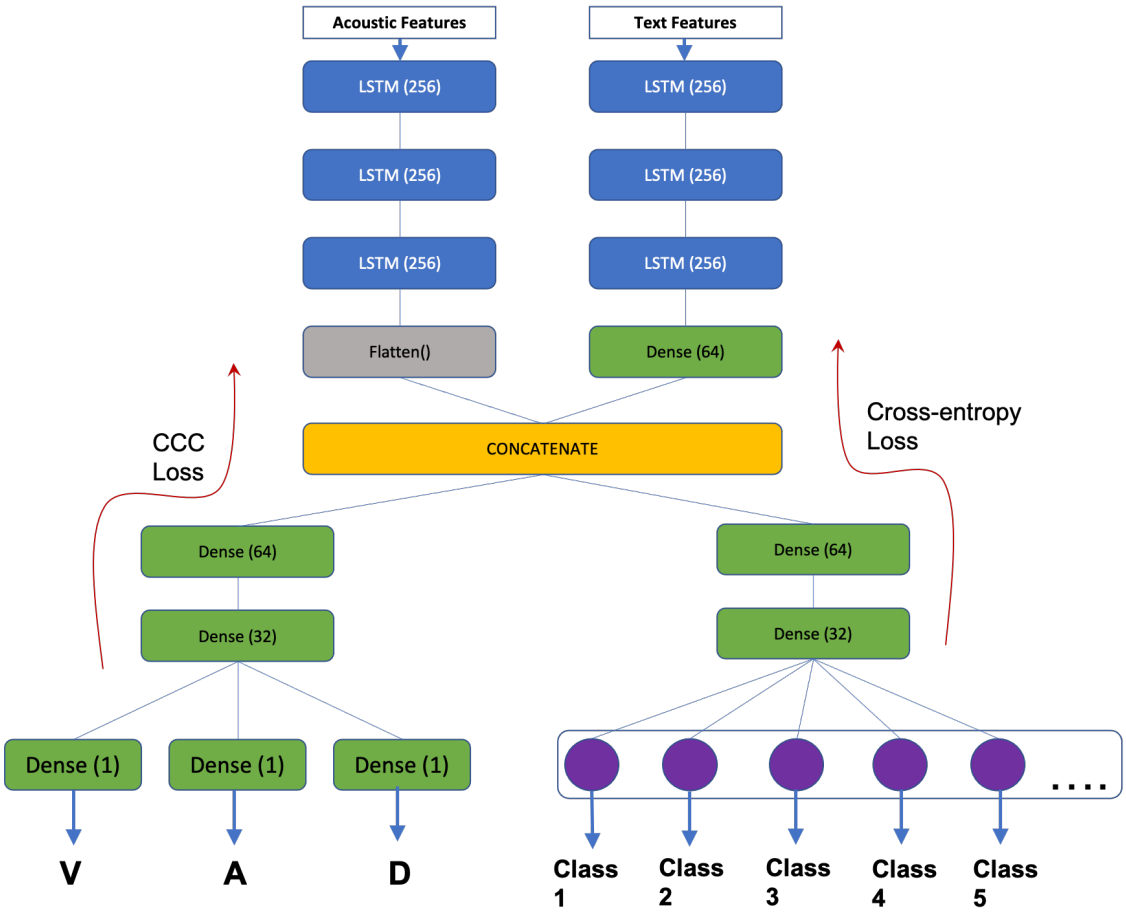


Fig. 3.19. Architecture of the parallel system using LSTMs for both the acoustic and text networks. V: Valence, A: Arousal, D: Dominance

This change in the definition of the problem to solve by the system not only affects thee architecture but also affects the configuration of the model. It mainly affects the loss function. Since the loss function is what makes a model learn a task, and the task has changed, the loss needs to change. The new loss function needs to model the learning in the previous regression task and in the new classification class. Therefore, the loss function is going to be a combination of the previous, $CCCL_{TOT}$ and Cross-Entropy Loss explained in 3.2.1. The $CCCL_{TOT}$ is applied to the output of original final dense layer and the new Cross-Entropy Loss is computed over the output of the new dense layer. Therefore, the new loss will be $CE - CCCL_{TOT}$ as shown in equation 3.39

$$CE - CCCL_{TOT} = \gamma CE + (1\text{-}\gamma)CCCL_{TOT} \tag{3.39}$$

where the best γ has been found to be 0.7.

This system targets the same class distribution as the presented in 3.2.1.

4. RESULTS

This chapter presents the results of the different experiments that were carried out. Each of the experiments resulted in some performance improvement or worsement. In any case, comments and explanations of such outcomes from an experiment are given. The chapter is divided in three main sections: one for the development of the core Multi-task Regression model, one for the Sequential System and the other for the Parallel System.

The most important results are in table 4.1, where it can be seen that the sequential system outperforms the parallel system in both CCC score (+2.58%) and accuracy score (+5.3%).

System	**CCC**	**Accuracy**
Sequential	0.8378 ± 0.0078	0.7502 ± 0.0159
Parallel	0.812 ± 0.009	0.697 ± 0.011

Table 4.1. RESULTS FOR EACH SYSTEM ON THE DIMENSIONAL EMOTION REGRESSION : VALENCE, AROUSAL, DOMINANCE; MEASURED BY CCC AND DISCRETE EMOTION CLASSIFICATION: ANGER, SADNESS, HAPPY, NEUTRAL, FEAR, SURPRISE; MEASURED BY ACCURACY

4.1. Multitask Regression Model (Core)

4.1.1. Baseline

The thesis needs some base model as an starting point. This is known as baseline model. This allows the thesis to compare new models discovered throughout the thesis research with the baseline model and see if the research is making any progress or improvements over the starting point.

In terms of data, the baseline uses for training and testing the IEMOCAP dataset from which it extracts pyAudioAnalysis HSF features, 68 features per audio, and uses GloVe word embeddings to process transcriptions, 300 dimensional vector per word. The target labels are valence, arousal and dominance in a scale of 1 to 5. Furthermore, it uses a fixed maximum length of 554 tokens for the transcriptions, hence, shorter transcriptions are right padded with zeros. With this approach the vocabulary size is of 3438 words.

With regards to the model's architecture it has two concatenated LSTM networks, acoustic and linguistic. The input is processed in batches of 256 samples. The acoustic network is fed with 68 features from each batch of audios with 1 timestep, $(256, 1, 68)$, then at each layer of the LSTM stack it outputs 256 dimension vectors, the vector from

the last LSTM layer is flattened with a dropout probability of 30% (0.3). The linguistic network is fed with 554 dimension vectors which go through an embedding layer creating a $(256, 554, 300)$ shape of the input. Then there is a 3 layer LSTM with 554 time steps and hidden size equal to 256 for each layer. Afterwards there is a linear layer with 64 output cells and a dropout probability of 30% (0.3). Finally, the concatenation network is fed with the concatenation of the output from the acoustic and linguistic network, hence, the input is of the shape $(256, 132)$. Then the output goes through 2 linear layers with outputs of 64 and 32 where the final output is dropped with a probability of 40% (0.4). Lastly the information is fed to the last 3 individual 1 cell layers.

Learning is performed with the $CCCL_{tot}$ loss function. This loss is weighted as follows: valence, usess alpha weight parameter; arousal, uses beta weight parameter and dominance uses 1- alpha - beta. As discovered in [50] the best values performance wise are alpha, 0.7 and beta, 0.2. The learning rate is 0.001 and the model trains during 25 epochs.

The CCC score of this baseline model is **0.5080 ± 0.0020** [50].

4.1.2. Experiments

In order to be able to compare models with the baseline and among them, ran experiments need to be as much deterministic as possible. Experiments are ran with fixed seeds for the random number generators to achieve results' reproducibility. In addition, K-Fold cross-validation is employed. This method is an iterative process that consist in randomly splitting up the data set into k subgroups of the same size, $k - 1$ groups are used to train the model and the left out group is used to test the model. This process is repeated k times using k different groups for testing. In this thesis $k = 10$ is selected. Finally, the results of the model are the average of the k iterations. Furthermore, in this thesis, 15% of training data is used as validation data for each epoch. On top of that, early stopping is employed which is a form of regularization used to avoid over fitting when training a learner with an iterative method [52]. Training starts with $best\_validation\_CCC = -\infty$, then at each epoch of training the CCC is stored if it is better than $best\_validation\_CCC$. Therefore, the training stops if the model's CCC does not improve for at least 0.0001 (delta) after 10 consecutive epochs (patience). Moreover, to make the implementation of the model less computationally expensive hidden layer repackaging was employed. Also, to avoid typical LSTM gradient vanishing, gradient clipping was introduced. A glance at the results can be seen in figure 4.1 and a more detailed description is given in the following sections and table 4.2.

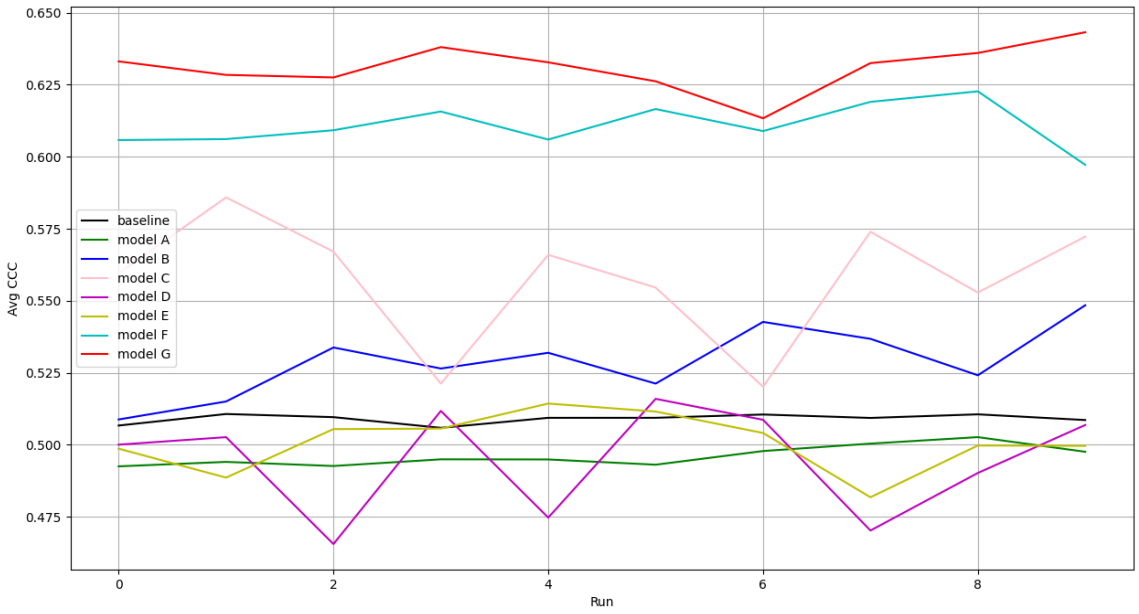


Fig. 4.1. Average test results of the regression on dimensional emotions

Name	V	A	D	Mean
Baseline	0.446 ± 0.0020	0.594 ± 0.0030	0.485 ± 0.0030	0.508 ± 0.0020
Model A	0.4549 ± 0.0444	0.5676 ± 0.0217	0.4699 ± 0.0153	0.4980 ± 0.0040
Model B	0.4170 ± 0.0188	0.6736 ± 0.0137	0.5135 ± 0.0100	0.5347 ± 0.0109
Model C	0.4686 ± 0.0355	0.6799 ± 0.0214	0.5248 ± 0.0200	0.5578 ± 0.0251
Model D	0.4312 ± 0.0078	0.5253 ± 0.0187	0.4466 ± 0.0222	0.4677 ± 0.1047
Model E	0.4164 ± 0.0444	0.6171 ± 0.0204	0.4935 ± 0.0175	0.5090 ± 0.0153
Model F	0.5928 ± 0.0221	0.6923 ± 0.0051	0.5503 ± 0.0096	0.6118 ± 0.0083
Model G	**0.6062 ± 0.0254**	**0.7093 ± 0.0094**	**0.5708 ± 0.0125**	**0.6300 ± 0.0142**

Table 4.2. REGRESSION CCC RESULTS

Model A: Adapted hidden size

The baseline model increases dimensionality in all of its LSTM layers and this makes the data more difficult to model most of the times, this phenomenon is known as *curse of dimensionlaity*. The baseline increases the dimensions of the acoustic features from 68 to 256 and also text features go from one integer word token to 300 dimension vector.

Therefore, the first approach of the thesis is to change this by adapting the layer cells to the dimensions of the input and actually reducing dimensions layer by layer in a hope of reducing the data's complexity having less but more meaningful vectors at each stage. The configuration for model A is the same of the baseline and only the hidden layer dimensions from LSTM stacks are modified. The acoustic network is fed with 68 features therefore hidden sizes for the 3 layers are $(50, 25, 25)$ respectively. The text network is left unchanges since the enlargement of dimensions that the embedding layer provides is

assumed to actually be benefitial for the model since it represents semantic relationships among words. However, this approach did not meet expectations since the average CCC score obtained over the test sets is **1% lower than the baseline, 0.4980 ± 0.0040**. There is no absolute truth about why this happens since deep learning models act as a black box, however, it is probable that the input space of the acoustic network is not complex enough to establish relationships between input and output and, hence, reducing the dimensions in the LSTM stack only makes this issue worse.

Model B: Adapted text sequences length

The baseline model process text very poorly. For example, it just takes the longest transcription in the data set and it padds with zeros all the other token sequences in order to have a fixed input size of the linguistic network. However, this thesis performed some analysis over the transcriptions and it can be seen that the transcription's mean length is 70 ± 40. Therefore, most of the transcriptions are padded with 484 zeros, this can have terrible impact on the model since the information corresponding to the 70 tokens of a transscription is lost in the fog of 484 padded zeros when the tokens traverse the LSTM cells time step by time step. This information lost can arguably affect the model since less information means worse learning and less generalization.

In order to overcome this issue, model B keeps the same configuration as the baseline but only changes the time step the linguistic LSTM takes and the mean of token sequence length plus standard deviation, 110. This change in the model has a positive impact on the model's performance, achieving an average of **0.5347 ± 0.0109** which is **2.67% better than the baseline**. This result shows has the thesis assumptions are right and a lot information is indeed lost inside the sea of zeros the previous padding pattern was causing. It is also true that some information coming from the outlier transcriptions that are longer than 110 tokens is lost however the information gain from the most common examples that before was lost makes up for it.

Model C: Text normalization

Following the same path as model B, more in depth analysis is done over the linguistic network. This time many of the steps for text normalization explained in 3.1.3 are employed. First, all contractions are split up since for example *"can't"* and *"can"* should not give the same information to the model, *"can't"* becomes *"can not"* the negation of can. Afterwards all words are transformed to lower case and lemmatization is performed over them. Lastly, the words belonging to a personalized list of stop words, punctuation and non alphanumeric words are removed. This normalization process cleans a lot the corpus of transcriptions and leaves the data set with the longest transcription token sequence from length 554 to 73. In addition it reduces the vocabulary from 3438 words to 2913 words.

All token sequences are fixed to length 73, right padding with zeros the shorter ones. Therefore, the model linguistic LSTM stack now has 73 time steps and everything else stays the same as in the baseline. This change in the model really pumps up the CCC

score, obtaining an average of **0.5578 ± 0.0251** which in the best case is **7.29% better than the baseline**. This results are due to the fact that the normalization process cleans very well the corpus leaving only the most relevant words which have the most information. The fact that the vocabulary size drops 15% allows makes the model simpler and transcriptions content shorter but more informative makes the model catch more important data relationships which translates into a better performance.

Stop words
can, again, each, too, our, any, nor,
only, why, was, out, other, now, doing, just,
ours, did, down, they, be, up, most, off, does,
are, were, having, do, has, not, until, before,
yourself, both, you, here, than, will, more,
because, once, where, when, your, how, same, few,
there, them, all, who, him, but, we, after

Table 4.3. CUSTOM SELECTION TO NOT USE AS STOP WORDS IN THE NORMALIZATION PROCESS

Model D: Low Level Acoustic Descriptors

This time the approach is to see if the model performs better using the LLD features from the pyAudioAnalysis feature set. The use of LLD features changes the input to the acoustic network to (256, 100, 34) where 100 is the number of frames per audio. Since the input changes, the acoustic LSTM stack is adapted to have 100 time steps whith 34 features each.

The results from this method are very poor, the model achieves an average CCC of **0.4677 ± 0.1047**. This results confirm what the authors of [50] showed, HSF features work better for this task. The fact that LLDs add a lot more of information to the input and dimensionality also increases a lot results in worse performance of the model since it is harder for it to stablish relationships among the data in such a high dimensional space. This increase of information is likely too much for the model due to its rather small architecture and perhaps a deeper architecture (i.e, deep CNN) with 30+ layers or longer training is able to capture the information correctly when LLDs are in use. However, for this thesis the computational resources and time were not available to experiment with such experiments.

Model E: eGEMAPS Acoustic Features

Model E is essentially the same as the baseline but instead of using pyAudioAnalysis HSF features, it uses eGEMAPS HSF features. By using this feature set the input size changes from (256, 68) to (256, 50). This results in a change of the acoustic network's first LSTM layer to have 50 cells.

The results are not very promising. The CCC score is **0.5090 ± 0.0153** which is **only**

1% better than the baseline. This experiment shows that eGEMAPS feature set is not as informative for emotion recognition as the pyAudioAnalysis set, most likely due to the fact that there are less features in the eGEMAPS set.

Model F: Model C + ComParE 2016 acoustic features

This time model C is used replacing the pyAudioAnalysis feature set with the ComParE 2016 HSF set to create model F. Model F changes the acoustic architecture to adapt to the new input size (256, 130) due to the size of the new set.

The results using this feature set are very promising. The CCC score skyrokects into **0.6118 ± 0.0083** which in the **best case is 11.01% better than the baseline**. This result demonstrates how the ComParE 2016 set is undoubtedly more informative for emotion recognition than eGEMAPS and pyAudioAnalysis sets. The used feature set is a newer developed feature set in comparison to the other two and hence, not only its larger number of features but also the quality of its features gives an outstanding performance.

Model G: Model F + combination of pAA, ComParE 2016 and Gaussian triad

Model G is just a small permutation of model F. Model F is employed with the supplementation of features from pyAudioAnalysis set that are missing in the ComParE 2016 set in order to combine the power of both feature sets. In addition, Gaussian triad features are also added. Therefore the acoustic network has to, once again, be adapted to the new input size which is 205 input features: 68 from pAA, 130 from ComParE 2016 and 7 from the Gaussian triads.

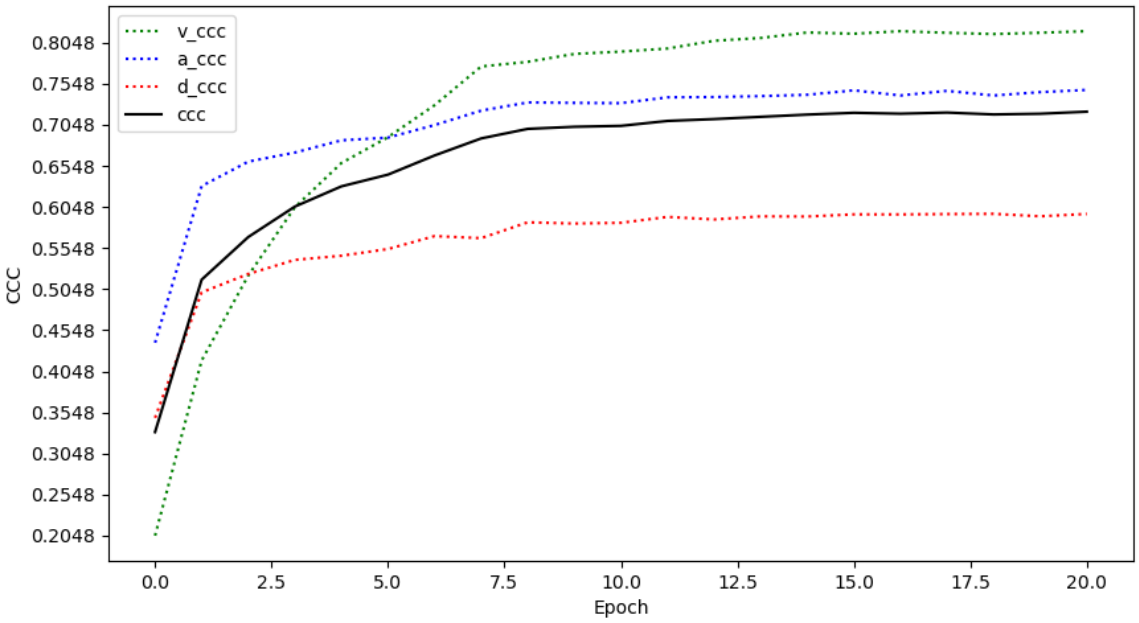


Fig. 4.2. CCC Results of the training of Model G on IEMOCAP data set.

The results are again very good and outperform model F. The CCC score grows up to **0.6300 ± 0.0142** which in the **best case is 13.42% better than the baseline**. This results shows how if pAA, ComParE 2016 feature sets and Gaussian triad features work

together they can cover each other weaknesses to produce a better model. Moreover, in table 4.2 it can be seen how to the model outperforms the baseline in all dimensions, 9% better in dominance dimension, 11% better on the arousal dimension and most importantly 16% in the valence dimension. This impressive improvement in valence is due to that the processing of text leaves a more informative corpus to the model. In natural language processing, text provides good information about sentiment (positive or negative), i.e valence dimension. Therefore, since the processed corpus is simpler, the text features extracted are better for the model and the valence results improve.

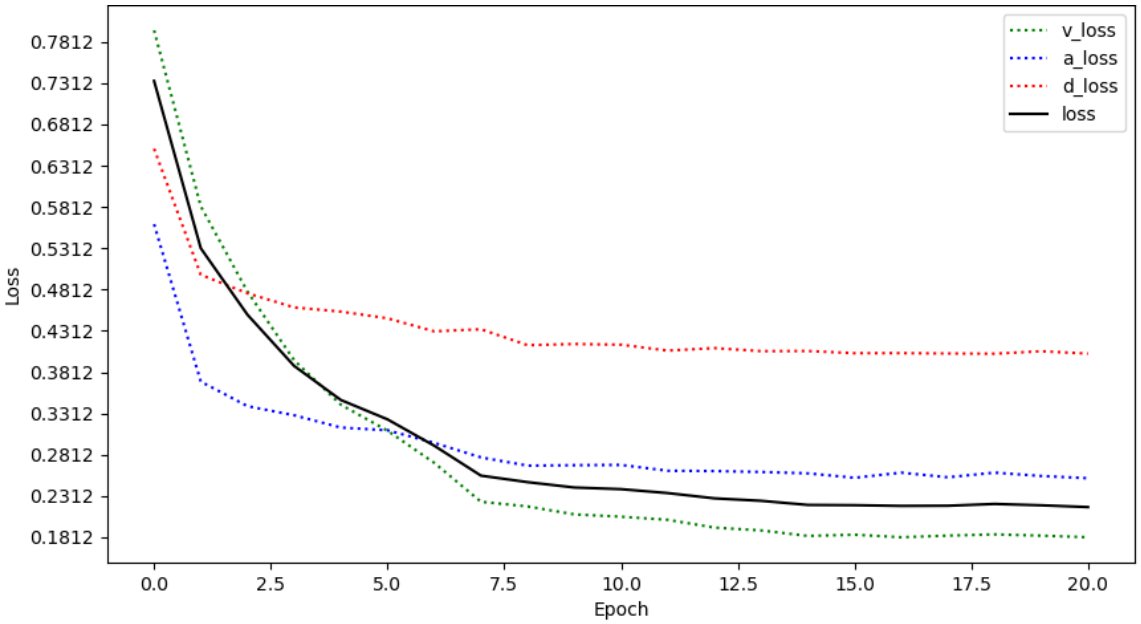


Fig. 4.3. Loss results of the training of Model G on IEMOCAP data set.

This confirms the findings of [35] where text features is shown to contribute most to valence dimension and acoustic features contribute more to the arousal and dominance dimensions. In figures 4.2 and 4.3 the training of this model can be seen.

4.2. Sequential System

4.2.1. Dimensional Emotion Regression (Step 1)

Baseline

The baseline this time is the same than in 4.1.1, with a CCC mean score of 0.508. The idea is that the developed multi-task regression model is adapted to work inside the sequential system. Due to the imbalances in discrete emotions as explained in 3.2.1 the number of data points is reduced from 10,038 to 5,638. **Experiments**

The experiments are run exactly the same as in the multi-task regression model (4.1.2).

As expected the cleaning of the data boosts the performance of the model in the task

since there is less amount and more easily separable data which allows the model to focus better.

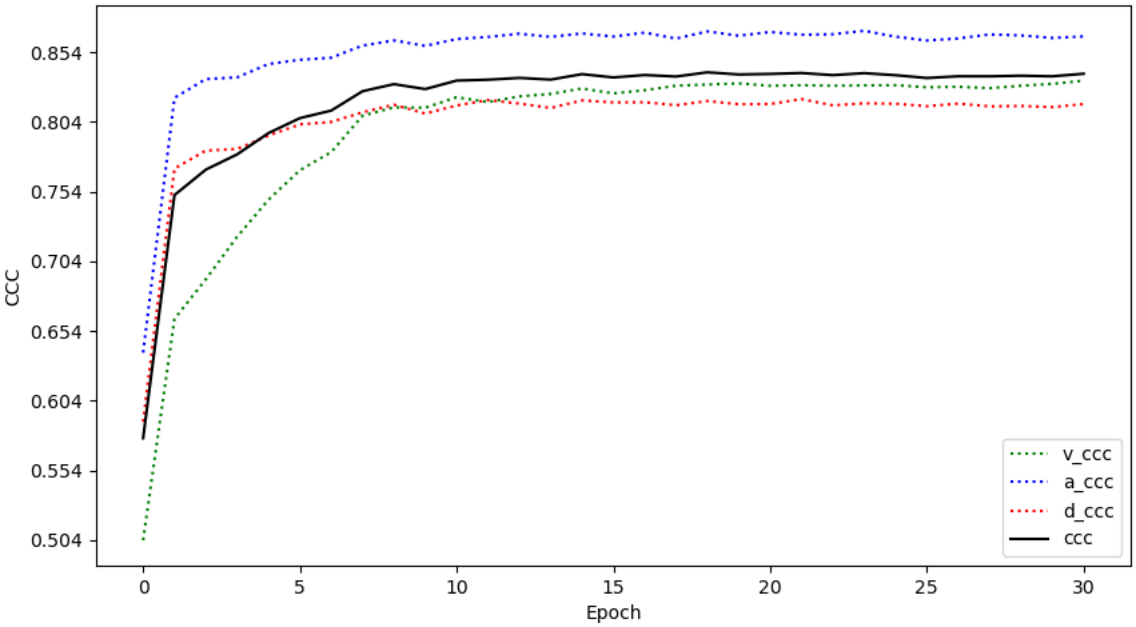


Fig. 4.4. Results of the CCC score for the training of one of the runs of the sequential system on the dimensional emotion regression task.

The training of the model can be seen in figures 4.4 and 4.5 where it achieves a CCC score of up to 0.84. However, in the testing data it gets a lower CCC score of **0.8378 ± 0.0078** which is the usual case in these models. The increase of performance over the baseline is 33%. However, this is not statistically comparable since this model is being trained on easier data to model.

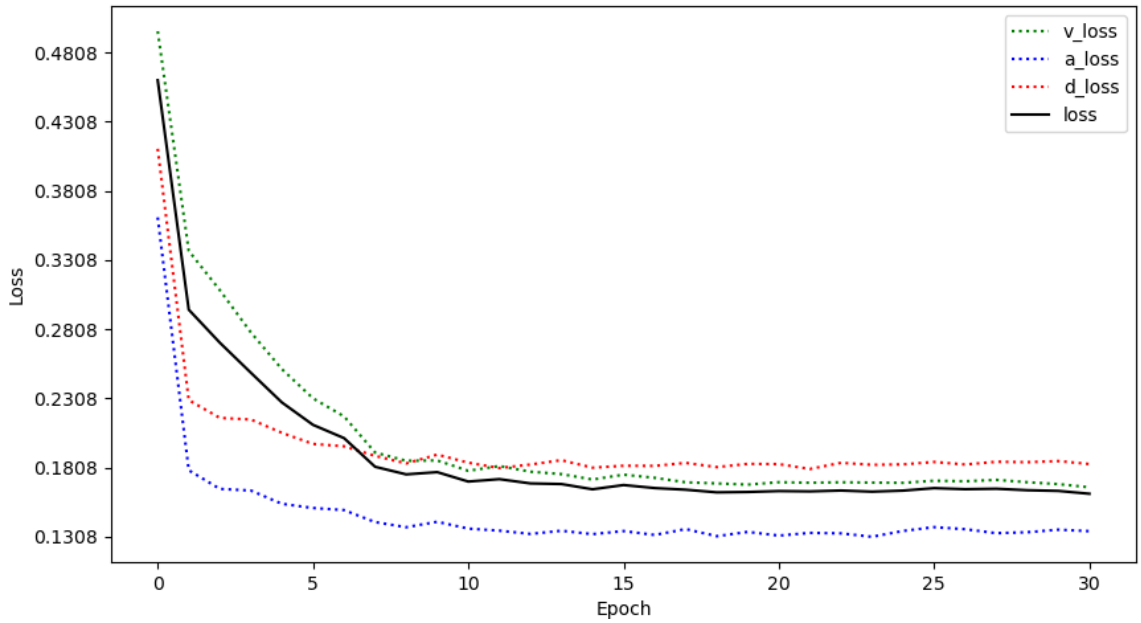


Fig. 4.5. Results of the Loss for the training of one of the runs of the sequential system on the dimensional emotion regression task.

4.2.2. Multi-class Emotion Classification (Step 2)

Baseline

Although now the problem is no more a regression problem, still a baseline accuracy is needed to be able to measure the quality of the models. In this work, the baseline accuracy is divided into 3 regions which are split by 2 thresholds.

Model	Accuracy	Precision	Recall	F1
Baseline	0.3008	-	-	-
XGB	0.7492 ± 0.0116	0.7413 ± 0.0174	0.7506 ± 0.0134	0.7415 ± 0.0151
Random Forest	0.7462 ± 0.0162	0.7371 ± 0.0175	0.7476 ± 0.0126	0.7385 ± 0.0140
Naive Bayes	0.7201 ± 0.0189	0.7290 ± 0.0175	0.7207 ± 0.0177	0.7121 ± 0.0178
MLP1	0.7428 ± 0.0151	0.7415 ± 0.0175	0.7494 ± 0.0173	0.7404 ± 0.0173
MLP2	0.7426 ± 0.0143	0.7399 ± 0.0146	0.7450 ± 0.0123	0.7364 ± 0.0139
KNN	0.7132 ± 0.0177	0.7047 ± 0.0171	0.7179 ± 0.0158	0.7082 ± 0.0167
SVM1	0.7492 ± 0.0155	0.7408 ± 0.0170	0.7504 ± 0.0139	0.7347 ± 0.0149
SVM2	0.7499 ± 0.0158	0.7465 ± 0.0181	0.7504 ± 0.0141	0.7349 ± 0.0159
SVM3	**0.7502 ± 0.0159**	**0.7481 ± 0.0179**	**0.7515 ± 0.0136**	**0.7361 ± 0.0151**
			Weighted	

Table 4.4. CLASSIFICATION RESULTS OVER ANGER, HAPPINESS, SADNESS, NEUTRAL, FEAR AND SURPRISE OF THE SEQUENTIAL SYSTEM.

The lower threshold is the accuracy given by the accuracy of the random rate classifier whose correct predictions are proportional to the distribution of classes among samples. The upper threshold is the accuracy of a zero rate classifier on the problem which always predicts the majority class [53]. In this specific case, the random rate classifier (RRC) has an accuracy of 24.80% and the zero rate classifier (ZRC) has an accuracy of 30.08%, as it can be seen in equations 4.2 and 4.1.

$$Accuracy_{ZRC} = \% \text{ of the majority class} = \% \text{ of neutral emotion samples} = 30.08\% \tag{4.1}$$

$$Accuracy_{RRC} = \sum_{i=1}^{\#\text{ classes}} \left(\frac{\#\ class_i \text{ samples}}{\text{total } \#\text{ samples}}\right)^2 * 100 = (0.1943^2 + 0.2881^2 + 0.1909^2 + 0.3008^2 + 0.007^2 + 0.0188^2) * 100 = 24.80\% \tag{4.2}$$

where the distribution of classes is the one after the cleaning proposed in section 3.2.1.

Therefore the regions would be region 1; accuracy lower than the random classifier and means awful model, region 2; accuracy in between random and zero rate classier

which means that the model adds value to the classification task, region 3; accuracy is higher than zero rate classifier which means the model starts to be good to be used as a predictor. All developed models significantly outperform the accuracy of region 3, being the SVM with C=10 the best model with a mean accuracy of 0.7502 as it can be seen in table 4.4.

Experiments

All experiments undergo a grid search process. This process looks for the best hyperparameters for each of the models. In order to evaluate how a set of hyperparameters performs on a model, the grid search runs a 10-fold cross validation on the model and takes the average accuracy score of the 10 independent runs. After that, the best performing model is chosen for the system to implement. A glance at the results can be seen in figure 4.6 and with more detail in the following sections and table 4.4.

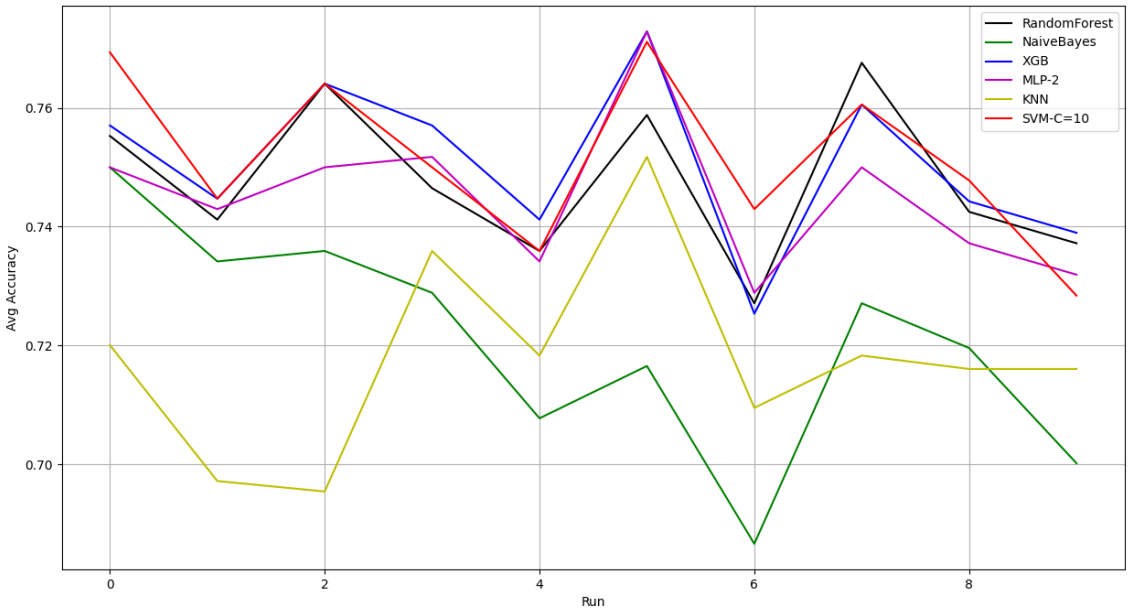


Fig. 4.6. Results of the discrete emotions classification on the training set.

eXtreme Gradient Boosting (XGB)

The XGB uses as the objective learner function the 'multi:softprob' which is a softmax function applied to the output. The booster employed is a gradient boosted tree. The learning rate of the classsifier is 0.3 and the maximum depth of the tree is 6, for each child the minimum sum of instance weight to keep partitioning is 1. When in a leaf node, the minimum loss or gamma required to make a further partition is 0. The total number of gradient boosted trees employed is 100 with a L2 regularization of 1.

The average accuracy obtained by this model in the 10-fold cross validation is **0.7492 ± 0.0116**.

Random Forest

The employed Random Forest uses 100 decision trees in its forest. This RF in order to measure the quality of a split uses the Gini impurity. The internal nodes need to at least

made up by two samples in order to be split and the leaf nodes need to be of size 1 at least to be considered as valid. Furthermore, in order to build the trees, bootstrap samples are used instead of using the whole data set.

The average accuracy obtained by this classifier is **0.7462 ± 0.0162**. **Naive Bayes**

This naive bayes approach implements Gaussian Naive Bayes Classifier that uses no prior probabilities of the classes and uses for variance smoothing 10^{-9}.

The average accuracy obtained by this classifier is **0.7201 ± 0.0189**.

Multi-Layer Perceptron

Two MLP Classifiers are employed, which only differ on the hidden sizes. They share the activation function between layers, ReLU. They use as the optimizer the ADAM algorithm with $\beta_1 = 0.9, \beta_2 = 0.999$ and $\epsilon = 10^{-8}$. L2 regularization is employed with an $\alpha = 0.0001$. The batch size employed is 200 and the learning rate is 0.001. Furthermore, the training runs for a maximum of 1000 epoch with a counter that increments each epoch in which the loss value does not improve for at least *tolerance* or 0.0001. When this counter reaches 10, training stop.

The first MLP Classifier uses 3 hidden layers of size: (256, 512, 256), the average accuracy achieved is **0.7428 ± 0.0151**.

The second MLP Classifier uses 5 hidden layers of size: (256, 512, 1024, 512, 256), the average accuracy achieved is **0.7426 ± 0.0143**.

K-Nearest Neighbors

The K-Nearest Neighbors Classifier employed uses the KDTree algorithm with a leaf size of 30 to compute the nearest neighbors. The metric used in the algorithm is the Euclidean metric. Finally, 5 neighbors are taken into account in the kneighbors queries and all of these points are weighted uniformly.

The average accuracy obtained by this classifier is **0.7132 ± 0.0177**.

Support Vector Machines

Thre SVM Classifiers are ran which only differ on the C or regularization parameter. They both employ a radial basis function kernel.

The first SVM Classifier uses $C = 1$ and achieves an average accuracy of **0.7492 ± 0.0155**.

The second SVM Classifier uses $C = 5$ and achieves an average accuracy of **0.7499 ± 0.0158**.

The third SVM Classifier uses $C = 10$ and achieves an average accuracy of **0.7502 ± 0.0159**.

As it can be seen in table 4.4, the SVM with a regularization factor of C = 10 is the best classifier in terms of accuracy found for the sequential system. Therefore, this model is the chosen one to use in conjunction with the regression model.

The confusion matrix of this model is shown in table 4.5. This matrix shows how the most confused emotions are neutral with sadness and vice-versa. This is related to their

Prediction	Ground Truth						Precision	Recall
	Anger	Happiness	Sadness	Neutral	Fear	Surprise		
Anger	164	1	22	6	0	1	0.85	0.74
Happiness	2	309	2	25	0	4	0.9	0.93
Sadness	12	0	98	10	3	2	0.78	0.50
Neutral	44	23	73	312	2	9	0.67	0.88
Fear	0	0	0	0	1	0	1	0.11
Surprise	0	0	0	0	0	0	0	0
Samples	222	333	195	353	6	16	0.79	0.78
	1125						Weighted	

Table 4.5. CONFUSION MATRIX OF SVM CLASSIFIER WITH C = 10

closeness in the dimensional space. In general, neutral is the emotion that is confused the most, since its positioned in the center of the valence-arousal-dominance space, its closer to the rest of emotions and it is more easy to misclassify. Moreover, due to the lack of samples for the fear and surprise emotions, the performance on these classes is awful as it can be seen in the matrix. However, in contrast to the parallel model, this model does correctly classify at least one sample as Fear.

4.3. Parallel System: Multitask regression and classification

Baseline

The baseline design for this model is going to be exactly the same as the one describe in 4.1.1 with the presented change of the new dense layer branch commented in 3.2.2 after the concatenation layer in order to adapt it to perform both regression and classification in parallel.

The input is still going to be IEMOCAP data which feeds 68 features per audio and 300 dimensional vectors per word into the network. However, the output changes from 3 variables to 9 variables. This output is formed by the valence, arousal, dominance, and a output vector with six elemtnss in which each index i represents the probability of the sample to be of $class_i$. Another relevant change is the loss function that now is the $CE - CCL_{TOT}$, see equation 3.39.

Finally the baseline values are a **CCC score of 0.508** (4.1.1) and **Accuracy of 0.3008** (Zero Rate Classifier, 4.2.2). The selected model will be the one with the greatest accuracy over the set of 6 emotions and greatest CCC on the dimensional emotion space.

Experiments

The experiments are ran following the same guidelines as the presented in section 4.1.2. However, this time the value used for early stopping is combined loss $CE-CCL_{TOT}$.

All the models proposed in this section outperform the selected baselines as it can be seen in table 4.6.

Model	Accuracy	Weighted Precision	Weighted Recall	Weighted F1	CCC
Model A	0.604 ± 0.013	0.634 ± 0.021	0.640 ± 0.002	0.639 ± 0.01	0.773 ± 0.014
Model B	**0.697 ± 0.011**	0.710 ± 0.003	0.706 ± 0.023	0.709 ± 0.012	**0.812 ± 0.009**

Table 4.6. CLASSIFICATION AND REGRESSION RESULTS OVER DISCRETE EMOTIONS(ANGER, HAPPINESS, SADNESS, NEUTRAL, FEAR AND SURPRISE) AND DIMENSIONAL EMOTIONS (VALENCE, AROUSAL, DOMINANCE) OF THE PARALLEL SYSTEM.

Model A: Adapted baseline

This model is the baseline architecture with the changes presented above, section 4.3, in order to adapt it to a classification and regression task.

The obtained accuracy is **0.6045 ± 0.0131**, which is 30% better than the accuracy baseline. The CCC **0.7739 ± 0.0143**, which is 14.4% better than the CCC baseline. However, in the case of CCC it is not statistically comparable to the baseline since this model is being trained on easier data to model.

Model B: Multi-task regression model G adapted

Model B takes the findings from the regression model section 4.1. This model implements the changes to the baseline, i.e transcriptions normalization, adjustment of transcription lengths, combination of acoustic features and more which make model G provide a significantly better performance. Basically, model B is the same as model G (4.1.2) but with the changes to adapt it to do both classification and regression.

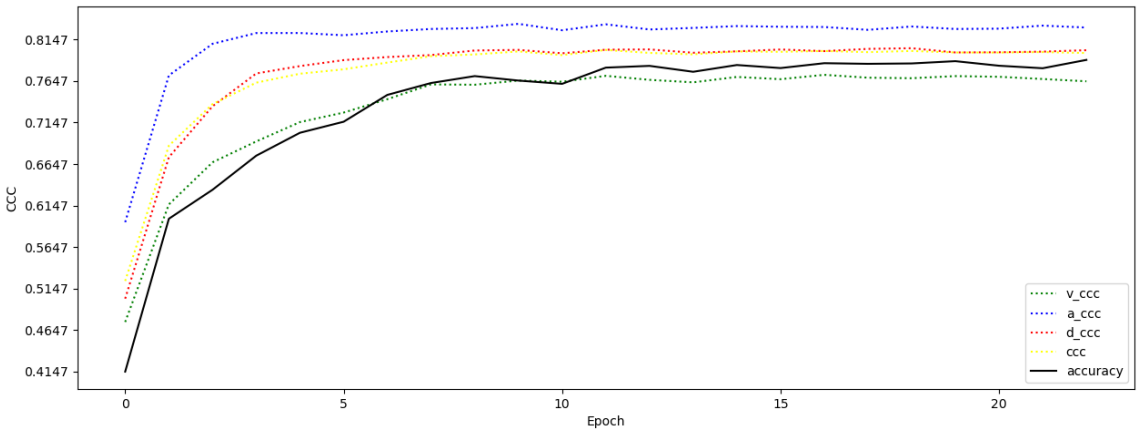


Fig. 4.7. Results of the CCC score for the training of one of the runs of the parallel system on the dimensional emotion regression task.

The training of this model achieves an accuracy of up to 0.76 as seen in figures 4.7

and 4.8. However, during testing obtained mean accuracy is **0.6972 ± 0.0113**, which in the best case is **39.64% better than the accuracy baseline**, and CCC **0.8123 ± 0.0098**, **30.04% better than CCC baseline**. However, in the case of CCC it is not statistically comparable since this model is being trained on easier data to model. This is the best model accuracy and CCC wise and hence is the selected one for this system.

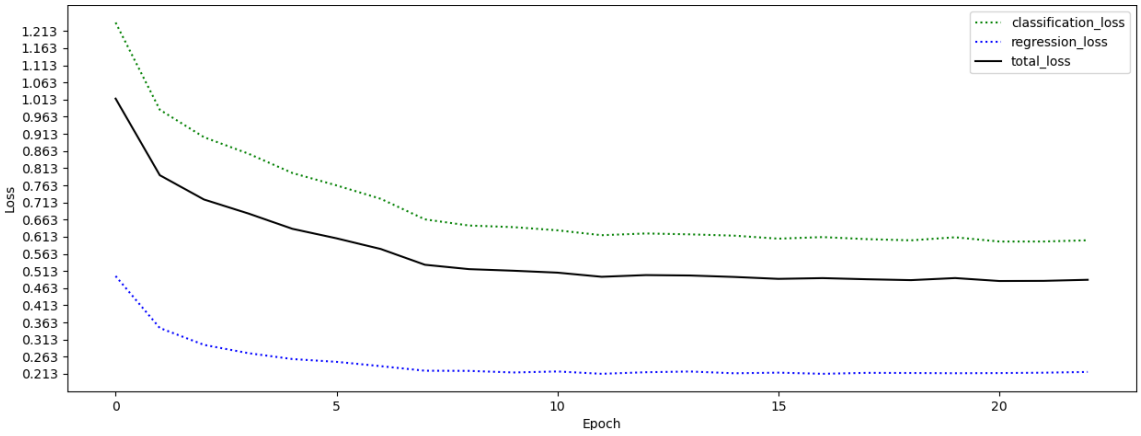


Fig. 4.8. Results of the Loss for the training of one of the runs of the parallel system on the dimensional emotion regression task.

As it can be seen in the confusion matrix of table 4.7 the neutral emotion is again the most confusing emotion to model, similar to what happened in the sequential model. In addition, sadness and neutral emotions are difficult for the model to distinguish, also similar to the sequential model. The performance is again very bad in fear and surprise emotions due to the little samples from those classes, but in contrast to the sequential model, this time there is not even one classification for the fear emotion.

Prediction	**Ground Truth**						**Precision**	**Recall**
	Anger	**Happiness**	**Sadness**	**Neutral**	**Fear**	**Surprise**		
Anger	184	10	3	20	1	11	0.8	0.75
Happiness	4	232	7	16	1	1	0.89	0.76
Sadness	5	17	147	58	1	2	0.64	0.67
Neutral	53	45	64	244	3	8	0.59	0.72
Fear	0	0	0	0	0	0	0	0
Surprise	0	0	0	0	0	0	0	0
Samples	256	304	221	338	6	22	0.71	0.71
	1147						Weighted	

Table 4.7. CONFUSION MATRIX OF THE PARALLEL SYSTEM DISCRETE EMOTION CLASSIFICATION.

4.4. Transfer Learning for the Sequential System

In this section the results obtained from applying transfer learning on the best performance system are presented. The sequential system is selected. Since transfer learning can only be applied to deep learning architectures, only the Multi-task Regression model (first step of system) will be taken into account. The employed data sets are IEMOCAP [35] and MSP [36]. Transfer learning will be conducted in both directions, from IECMOAP data set to MSP data set and vice-versa.

The model used in the process is model G, section 4.1.2.However, in order to apply transfer learning between models, some changes have to be done to the architectures. This is due to the fact that the two data sets are different and hence the vocabulary included in them changes, and the transcription lengths also differ from one another. The MSP data set has a larger corpus with a vocabulary size of 26,590 words and a maximum transcription length after normalization of 41 tokens. On the other hand, the IEMOCAP data set is smaller with a vocabulary composed by 2,913 words and a maximum length of normalized transcriptions of 73 tokens. These characteristics affect the size of the embeddings matrix used by the model and the time steps employed by the LSTM when processing text. Therefore, in order to not lose any information when processing data sets, the thesis adapts the architecture so that the embeddings matrix uses the vocabulary from the MSP data set (26,590 words) and the time steps, average text length, is chosen to be 73 as in the IEMOCAP data set. This vocabulary choice can seem harmful in terms of information, since at first glance the vocabulary of the IEMOCAP data set is not included in the model. However, the thesis investigates on this issue and finds that 2649 words from the IEMOCAP's vocabulary are in the MSP's vocabulary, hence, only 9% of the IEMOCAP vocabulary is lost with this approach.

The pre-trained model can transfer all of its learning, weights and biases, to the new model or only parts of it. Six different approaches are taken: transferring the total learning, only the embedding layer weights, only the acoustic network, only the dense layers previous to the output (Concatenation Network), only the linguistic network(which includes embedding layer) and transferring all but the 3 dense output layers.

Pre-trained component	CCC
None	0.6300 ± 0.0142
Whole Network	0.6151 ± 0.0090
Whole Network w/o output layer	0.6202 ± 0.0127
Acoustic Network	**0.6307 ± 0.0230**
Concatenation Network	0.6231 ± 0.0055
Linguistic Network	0.6202 ± 0.0127
Embedding Layer	0.6196 ± 0.0119

Table 4.8. RESULTS OF TRANSFER LEARNING FROM MSP REGRESSION MODEL TO IEMOCAP REGRESSION MODEL

As it can be seen in both tables 4.8 and 4.9 transfer learning can help in this specific task. An improvement is seen on the IEMOCAP data set when transferring the learning from the acoustic network pre-trained on the MSP data set, a CCC score of 0.6307±0.0230 is obtained. This is an **improvement of 0.7%, or 2.30% in the best case**, over the previous best multi-task regression model, model G (CCC = 0.6300). Similarly, when transferring the learning in the acoustic network from the IEMOCAP data to MSP model, there is an **improvement from 0.4947 to 0.5216, 2.7%.**

Pre-trained component	CCC
None	0.4902 ± 0.0047
Whole Network	0.4815 ± 0.0142
Whole Network w/o output layer	0.4890 ± 0.0071
Acoustic Network	**0.5014 ± 0.0202**
Concatenation Network	0.4877 ± 0.0044
Linguistic Network	0.4883 ± 0.0099
Embedding Layer	0.4832 ± 0.0113

Table 4.9. RESULTS OF TRANSFER LEARNING FROM IEMOCAP REGRESSION MODEL TO MSP REGRESSION MODEL

5. CONCLUSIONS AND FUTURE WORK

As introduced in Chapter 1, the goal of this thesis is to develop a dimensional and discrete emotion recognition system. In order to accomplish this task, two different approaches are taken. The first approach is to build a sequential system and the second to build a parallel system. Both systems are evaluated on the IEMOCAP data set.

In order to build the systems, the thesis conducts a previous study on audio characteristics and its processing. Three different audio feature sets are evaluated: *pyAudioAnalysis, eGeMAPS, ComParE 2016*. In addition, the thesis also studies text linguistics and its processing.

The thesis first focus is to develop an architecture to best predict valence-arousal-dominance (dimensional emotion). The developed multi-task regression model with the proposed mixture of input acoustic features in 4.1.2 and proposed processing of input text is able to focus on the emotional data relationships better than the model proposed by Bagus Trisatmaja at Cambridge institution [50]. This is reflected in the performance boost of 13.42% in the mean CCC score achieved by this thesis' model. This multi-task regression model is then used as the core of both built systems

Afterwards, the sequential and parallel systems are designed. The sequential system operates in two steps. First, it extracts the dimensional emotion from speech using multi-task regression. Second, it classifies the the dimensional emotion representation into a discrete emotion using a state-of-the-art classifier. The parallel system in contrast to the sequential one, extracts both dimensional and discrete emotions in one step, using multi-task regression and classification at the same time.

After running all experiments, the thesis concludes that the sequential system outperforms the parallel system in classification, with an increase of 5.3% in accuracy over the parallel system. This result confirms that the psychological model of dimensional emotions proposed by J.A Russel and A.Mehrabian in 1977 [5] is correct. This psychological model states that any emotion can be represented in a three dimensional space conserving all of its meaning. Therefore, it is proven that it is a better approach to first map input dimensions to the three dimensional space of emotions proposed by [5] and then perform classification using these 3 variables as input. Another advantage of the sequential model is that since it operates on two steps it can focus on both tasks independently which makes it also outperform the parallel model in the regression of valence-arousal-dominance variable with a 2.6% increase in the CCC score. Therefore, the thesis selects the sequential system for emotion recognition.

In addition, the thesis also conducts research on whether transfer learning can be applied to this specific task. Transfer learning is applied to the selected sequential system in order to improve its performance. The IEMOCAP and MSP data sets are employed.

The thesis concludes that transfer learning is relevant on this task since it has improved performance in both data sets by 2.30% and 2.70%.

Finally, the following are some possible paths of research that could be followed in order to further develop and improve the system proposed by the thesis:

- **Multi-language approach:** One of the drawbacks of the presented system is the fact that it has only been analyzed over the English language. An interesting new approach will be to train the system on data bases where multiple languages co-exist and analyze its robustness extracting emotional information from different languages. This is a very hard task which nowadays is still unsolved with good performance due to the big generalization the model needs have.

- **Improve acoustic network:** Another drawback of the system is the performance in acoustic related dimensions (arousal and dominance). The performance is not comparable to the one for the text related dimension (valence). This is because the acoustic network does a moderate job at extracting information from acoustic features. One reason this is happening is because high statistical features are not informative enough, however in this thesis when the low level features are employed results are worse. The fact that results in this thesis worse when using low level features is likely not because these are not emotional informative but because the acoustic architecture is not deep or trained enough. In order to fix this, deep Convolutional Neural Networks could be used to process this large amounts of features. Another solution, is to train the network for longer since LSTMs take time to establish good relationships among data.

- **Emotional embeddings:** The performance of the system in text features is decent but has room to improve. The linguistic network uses an embedding layer to capture the semantic relationship between words. However, if the system is more customized for the specific task which is related with emotions, it makes more sense to use an embedding layer that allows the model to capture both semantic and emotional relationships among words. This embeddings are built using the existing emotional lexicon and combining it with existing word embeddings which results in a hybrid representation of semantics and emotion [54]. This approach will most-likely better the performance of the system, not only on the text-related dimension (valence) but also on arousal and dominance.

- **State-of-the-art language model:** More improvements can be made to the emotional information extracted from text. A change from them basic LSTM architecture of the linguistic network to a more complex such a BERT model or any transformer architecture would lead to significant improvements in performance. The use of architectures like transformers provide an attention mechanism that focuses better on the most relevant data relationships, which in the case of BERT is even more powerful since it operates bidirectionally on the sequences it receives.

6. SOCIOECONOMIC AND REGULATION

6.1. Socioeconomic Impact

In the last few years the transformation of society has come by the hands of technology, investigation and digitization. In addition, the TIC sector is having a big impact on the doings of governments and opening new opportunities to the most vulnerable.

It is difficult to quantify the benefits of the technological innovation. Many disruptive technologies are pushing out solutions to complex problems and are changing the ways things are done in all sectors and society aswell. Nowadays, 95% of global population lives in an area where there is mobile network coverage and more than 50% have access to the Internet. However, in 3rd world countries, the penetration in 2016 was only of 15%. Countries such as Tanzania, India, Bangladesh or Thailand have not achieved the digital goals set for year 2020 [55]. Spain keeps promoting the deployment of networks and the digital economy. Spain is one of the countries with most fiber coverage in the world, with an 80% of coverage versus the European average, 34% [56]. According to the study "Estado de Digitalización de las Empresass y Administraciones Públicas españolas" made by Vodafone; 30% of the big companies give great importance to the use of big data; nearly half of them, independently from their size, point out Internet of Things as an important service that they want to exploit; 42% of small and medium size companies look for integrating e-commerce in their business, and 1 out of 5 big companies use machine learning in their business [57].

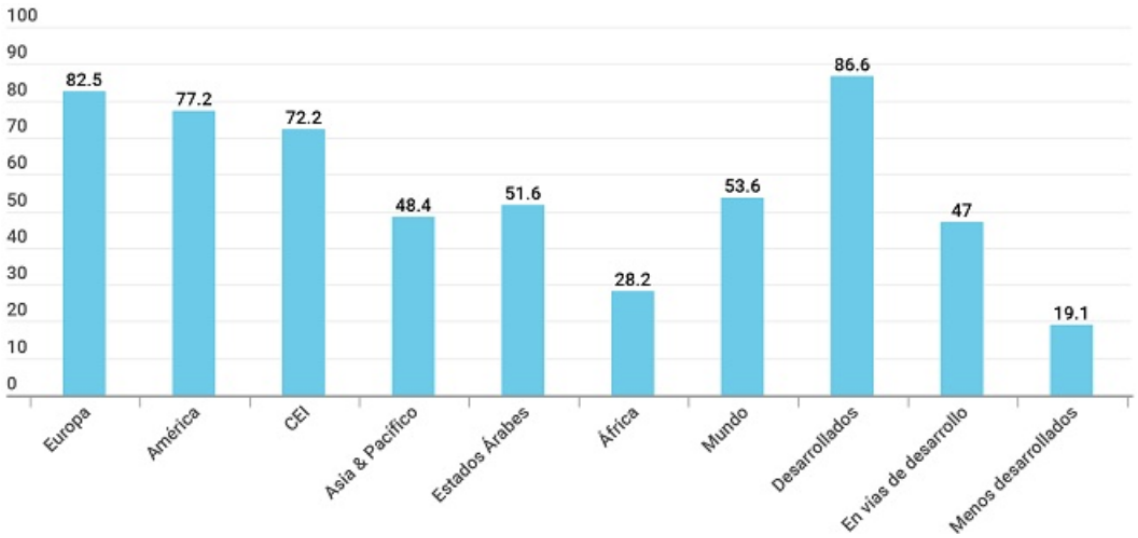


Fig. 6.1. Percentage of population with internet access, by region and development status

Undoubtedly, 2020/21 will be remembered as the year in which governments, com-

panies and citizens saw the necessity of changing their ways of living, thinking, working, learning, and especially communicating. The global emergency generated by the COVID-19 has been a great challenge for humanity. In a world where technology is part of our lives, connectivity has enabled the continuity of most businesses thanks to remote working or e-commerce, and in a few months things have evolved what would have taken 3 or 5 years in a normal situation. For businesses which used to be completely unrelated to technology, the Internet has been of great help to satisfy their need of recycling themselves. Sectors such as education and consulting have been able to keep serving their clients thanks to remote working, video calls, messaging apps and social network apps.

This crisis has open a wide range of possibilities for technological innovation since the generation of new technologies has been key for public and private corporations. The main focus right now is on the healthcare sector, seeking for technologies that allow us to quickly solve the pandemic with things such as ventilators, body protectors, thermal cameras or vaccines. However, for TIC companies, IoT is one of their big bets. It is estimated that in Spain, each citizen will have an average of 7 devices connected to the network thanks to ideas such as *Smart City* which is expected to improve well-being, stabilize climate change and optimize mobility. In addition, most innovation is focused on applications that improve security by controlling social distancing through private networks that warn users about their distance to others; motion and energy detectors; or the monitoring of infected people in quarantine with geolocation method to avoid spreading the virus.

The Digital European Agenda has set as a goal establishing a unique digital market, strengthen online security, promoting internet access to all countries to facilitate digital integration and investing in research and innovation. The main goals for 2020 were the reduction of fees in electronic communications , the development of high speed network for the main socioeconomic drivers thanks to the evolution of mobile networks and 5G, the agreement on a time plan for the commercial distribution of 5G, and free access to WiFi networks in public location thanks to WiFi4EU. In terms of privacy and personal data, there is a search for a better protection of user data by means of a new legislation about privacy and data protection with a new regulation for data protection [58] and thanks to the European Agency of Security of Networks and Information (ENISA) [59].

It is also worth mention the existance of a Digital Agenda for Spain, approved in 2013. In "España Digital 2025", which was urgently written due to the pandemic situation of COVID-19, the main goals guaranteeing a decent digital connection for 100% of the population, continue to lead Europe in the deployment of 5G networks, reinforce digital skills of the Spanish workforce, strengthen Spanish cybersecurity, promote the digitization of public administrations, speed up companies' digitization (specially "PYMES" and startups) and assuring the protection of consumer's data and privacy with the help of artificial intelligence. Nowadays, the supervision of this proposals is done by the Minister of Economic Affairs and Digital Transformation.

Finally, the use of big data in the corporation world is allowing for new opportunities

Fig. 6.2. Mechanism to monitor the location of an infected person by means of an smart wristband

of growth for new companies and specialized startups. It is possible to store data from different sources, process them and analyze them in real time to satisfy the customer's demands. Big tech companies such as Google, Amazon, IBM or Microsoft offer cloud service to deal with a companies data. Tools such as Google Cloud, Amazon Web Services, IBM-Cloudera or Microsoft Azure allow big companies to save on costs and time, understand markets, control the reputation of a company on the Internet, increase client retention and develop new products. All the mentioned have to be developed in secure environments, since a greater digitization of society means a greater exposure to cyber attacks.

The experiments shown in this thesis have applications on multiple fields. A system for emotion recognition through voice deployed *on-cloud* or *on-premise* can add a lot of value to fields such as neuromarketing, education or healthcare.

6.2. Project management and budget

This section unfolds the project management and budget, following the guidelines imposed in the UNE-ISO 21500:2013 [60] and the Project Management Body of Knowledge [61], crucial tool for the effective management of a project in any industry.

6.2.1. Management

The table 6.1 shows the hours put into each activity and the total number of hours for this thesis.

ID	Task	Hours
A	Search for a thesis subject and first meeting with professor	15
B	Study of the state of the art in the field of emotion classification: read papers and studies about classification methods, different feature sets, architectures, psychology, linguistics, data sets.	30
C	Reaching out to leaders of research groups in the thesis field from different institutions to obtain their developed data sets and sending them they signed license for research. Downloading the data and feature processing.	40
D	Development and implementation of sequential system with the corresponding evaluation and tuning of the models.	80
E	Development and implementation of parallel system with the corresponding evaluation and tuning of the models	20
F	Analysis of the obtained results with the corresponding comparison between experiments and study of performance	10
G	Thesis writing following the guidelines by University Carlos III of Madrid	100
H	Creation of the presentation and defending the thesis in front of the evaluation court	20
	Total	315

Table 6.1. PERFORMED TASKS AND DURATION

In table 6.2 the Gantt diagram is presented which shows the overview of the activities. As it can be seen in the Gant diagram, some tasks have been carried out simultaneously.

TASK	JANUARY	FEBRUARY	MARCH	APRIL	MAY	JUNE
A: Search of thesis subject and first meeting with tutor						
B: Study state-of-the-art in the field						
C: Gather databases from different institutions						
D: Development of sequential system						
E: Development of parallel system						
F: Analysis of the resultss						
G: Writing the thesis						
H: Create presentation for the thesis						

Table 6.2. GANT DIAGRAM.

6.2.2. Budget

The budget of the project has been calculated taking into account the materials cost, human resources and indirect costs which are detailed below.

The costs of material include all the resources employed in the development of experiments. In terms of software, Python was utilized which has an open-source license,

Overleaf which also has an open-source license was employed for the thesis writing. In terms of hardware, a MacBook Pro was employed. These costs can be seen in table 6.3.

The human resources costs are calculated taking into account the work hours put in by the people involved: Professor Antonio Artés, thesis supervisor, and the author, as the engineer. These costs can be seen in table 6.4

Product	**Cost**	**Redeemed cost (months)**	**Time of use (months)**	**Total cost**
MacBook Pro Retina 15"	1,500€	60	6	150€
			Total	150€

Table 6.3. COSTS OF MATERIAL

Name	**Charge**	**Salary (€/hour)**	**Hours**	**Total cost**
Professor Antonio Artés	Senior Engineer	45	10	450€
Martín Iglesias Goyanes	Junior Engineer	25	315	7,875€
			Total	8,325€

Table 6.4. HR COSTS

The indirect costs include the transport necessary to go to attend the university in order to meet the supervisor. The cost is very low and its just the the cost of two monthly subscriptions to the Madrid Transport Network, 40€, since most of the meeting were conducted only for convenience. The total cost also includes the percentage over costs for human resources and materials. In the projects carried out inside UCIIIM, the most common value is 15%. It can be seen in table 6.5 how the costs rise up to 9,792.25€.

Cost name	**Cost**
Cost of Material	150€
Cost of HR	8,325€
Indirect costs	40€
University percentage	1,277.25€
Total	9,792.25€

Table 6.5. TOTAL COSTS

6.3. Regulation

The Organic Law 3/2018, from December 5th, of Protection of Personal Data and guarantee of the digital rights is the main legislation in terms of data protection in Spain [62]. Adapts the Spanish legal system to the Regulation (UE) 2016/679 of the European Parliament and the European Council. and therefore the fundamental right of persons to personal data protection is imposed according to what is established by this organic law and in the mentioned regulation. In addition, the Organic Law 3/2018 guarantees the digital rights to the people following the mandate established in the article 18.4 of the Constitution.

Furthermore, the Spanish Agency for Data Protection (AEPD) is the corporation in charge of ensuring the compliance of the normative about data protection, informing citizens and advising them in the presentation of complaints related to data protection. It is a public entity, legal entity and it operates with independence of the other pubic entities. It cooperates with the government through the Ministry of Justice.

Intellectual property includes the set of rights related with literary, artistic or scientific works and the benefits obtained thanks to their creation. The Law of Intellectual Property is a Royal-Decree Law 1/1966, from 12th of April, an regulates the intellectual property [63].

In Europe, is worth mentioning the Regulation (UE) 2016/679 from the European Parliament and Council in April 27th, 2016. The rules in this regulation include the protection of people with respect to the processing of their data as a fundamental right, taking into account the rapid technological evolution and globalization , which have motivated the rise in the flowing of personal data inside the European Union [58].

The databases employed in this thesis have a Creative Commons license and they are open for non-commercial uses. The thesis has been written in Overleaf, which has an open source license and the experiments have been ran in Python, which has an open source license as well, *Python Software Foundation License*

BIBLIOGRAPHY

[1] M. F. Pradier, "Emotion recognition from speech signals and perception of music," M.S. thesis, Universität Stuttgart, Institut für Systemtheorie und Bildschirmtechnik Lehrstuhl für Systemtheorie und Signalverarbeitung., 2011.

[2] M. Guerri, *Que son las emociones?* Psicoactiva, Accessed: 14-06-2021, May 2017. [Online]. [Online]. Available: https://www.psicoactiva.com/blog/que-son-las-emociones/.

[3] M. Jalife, *Exponencial crecimiento de patentes de inteligencia artificial*, El Financiero, Accessed: 14-06-2021, Feb. 2019. [Online]. [Online]. Available: https://www.elfinanciero.com.mx/opinion/mauricio-jalife/exponencial-crecimiento-de-patentes-de-inteligencia-artificial/.

[4] C. D. von Eitzen., *Tecnologias exponenciales: Que son, cuales son y en que consisten: Ia, blockchain, nanotecnologia*, Blog de ChristianDvE, Accessed: 14-06-2021, Aug. 2019. [Online]. [Online]. Available: http://www.christiandve.com/2019/08/tecnologias-exponenciales-que-son-cuales-son-en-que-consisten/.

[5] J. A. Russell and A. Mehrabian, "Evidence for a three-factor theory of emotions," *Journal of Research in Personality*, vol. 11, no. 3, pp. 273–294, 1977. DOI: https://doi.org/10.1016/0092-6566(77)90037-X. [Online]. Available: https://www.sciencedirect.com/science/article/pii/009265667790037X.

[6] N. Morán, J. Pérez, and W. Rodriguez, "Reconocimiento de estados emocionales de personas mediante la voz utilizando algoritmos de aprendizaje de máquina," vol. 5, pp. 41–52, Dec. 2018.

[7] G. Fairbanks and W. Pronovost, "An experimental study of the pitch characteristics of the voice during the expression of emotion," *Speech Monographs*, vol. 6, no. 1, pp. 87–104, 1939. DOI: 10.1080/03637753909374863. eprint: https://doi.org/10.1080/03637753909374863. [Online]. Available: https://doi.org/10.1080/03637753909374863.

[8] C. Williams and K. Stevens, "Emotions and speech: Some acoustical correlates.," *The Journal of the Acoustical Society of America*, vol. 52 4, pp. 1238–50, 1972.

[9] R. Cowie *et al.*, "Emotion recognition in human-computer interaction," *Signal Processing Magazine, IEEE*, vol. 18, pp. 32–80, Feb. 2001. DOI: 10.1109/79.911197.

[10] W. JAMES, "II.—WHAT IS AN EMOTION ?" *Mind*, vol. os-IX, no. 34, pp. 188–205, Apr. 1884. DOI: 10.1093/mind/os-IX.34.188. eprint: https://academic.oup.com/mind/article-pdf/os-IX/34/188/9278514/os-IX_34_188.pdf. [Online]. Available: https://doi.org/10.1093/mind/os-IX.34.188.

[11] C. L. de Luis., *Los 3 componentes de las emociones*, La mente es maravillosa, Accessed: 14-06-2021, Jun. 2019. [Online]. [Online]. Available: https://lamenteesmaravillosa.com/los-3-componentes-las-emociones/.

[12] P. Ekman and H. Oster, "Facial expressions of emotion," *Annual Review of Psychology*, vol. 30, no. 1, pp. 527–554, 1979. DOI: 10.1146/annurev.ps.30.020179.002523. eprint: https://doi.org/10.1146/annurev.ps.30.020179.002523. [Online]. Available: https://doi.org/10.1146/annurev.ps.30.020179.002523.

[13] O. Mitruț, G. Moise, L. Petrescu, A. Moldoveanu, M. Leordeanu, and F. Moldoveanu, "Emotion classification based on biophysical signals and machine learning techniques," *Symmetry*, vol. 12, p. 21, Dec. 2019. DOI: 10.3390/sym12010021.

[14] J. Russell, "A circumplex model of affect," *Journal of Personality and Social Psychology*, vol. 39, pp. 1161–1178, Dec. 1980. DOI: 10.1037/h0077714.

[15] A. Mehrabian, "Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in temperament," *Current Psychology*, vol. 14, pp. 261–292, 1996.

[16] Wikipedia contributors, *Emotional prosody — Wikipedia, the free encyclopedia*, [Online; accessed 13-June-2021], 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Emotional_prosody&oldid=1022166436.

[17] S. Mohammad, "Obtaining reliable human ratings of valence, arousal, and dominance for 20,000 english words," Jan. 2018, pp. 174–184. DOI: 10.18653/v1/P18-1017.

[18] S. M. Mohammad, *Sentiment analysis: Automatically detecting valence, emotions, and other affectual states from text*, 2021. arXiv: 2005.11882 [cs.CL].

[19] C. Strapparava and A. Valitutti, "Wordnet-affect: An affective extension of wordnet," *Vol 4.*, vol. 4, Jan. 2004.

[20] *Dartmouth summer research project on artificial intelligence*, Accessed: 14-06-2021, 1960. [Online]. [Online]. Available: https://www.livinginternet.com/i/ii_ai.htm.

[21] H. Moravec, "The stanford cart and the cmu rover," in *Autonomous Robot Vehicles*, I. J. Cox and G. T. Wilfong, Eds., Springer-Verlag, Jul. 1990, pp. 407–41.

[22] H. Heidenreich, *What are the types of machine learning?* Accessed: 14-06-2021, Dec. 2018. [Online]. Available: https://towardsdatascience.com/what-are-the-types-of-machine-learning-e2b9e5d1756f.

[23] *Función de activación - redes neuronales*, Accessed: 14-06-2021, Dec. 2018. [Online]. Available: https://www.diegocalvo.es/funcion-de-activacion-redes-neuronales/#:~:text=Definici%C3%B3n%20de%20funci%C3%B3n%20de%20activaci%C3%B3n,o%20(%2D1%2C1)..

[24] J. Dellinger, *Weight initialization in neural networks: A journey from the basics to kaiming*, Accessed: 14-06-2021, Apr. 2019. [Online]. Available: https://towardsdatascience.com/weight-initialization-in-neural-networks-a-journey-from-the-basics-to-kaiming-954fb9b47c79.

[25] Accessed: 14-06-2021. [Online]. Available: http://introtodeeplearning.com/slides/6S191_MIT_DeepLearning_L1.pdf.

[26] *Is transfer learning the final step for enabling ai in aviation?* Jan. 2021. [Online]. Available: https://datascience.aero/transfer-learning-aviation/.

[27] A. Cutler, D. Cutler, and J. Stevens, "Random forests," in. Jan. 2011, vol. 45, pp. 157–176. DOI: 10.1007/978-1-4419-9326-7_5.

[28] V. Zhou, *A simple explanation of gini impurity*, Accessed: 14-06-2021, Mar. 2019. [Online]. Available: https://victorzhou.com/blog/gini-impurity/.

[29] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[30] T. Yiu, *Understanding random forest*, Accessed: 06-05-2021, Aug. 2019. [Online]. Available: https://towardsdatascience.com/understanding-random-forest-58381e0602d2.

[31] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16, San Francisco, California, USA: ACM, 2016, pp. 785–794. DOI: 10.1145/2939672.2939785. [Online]. Available: http://doi.acm.org/10.1145/2939672.2939785.

[32] Wikipedia contributors, *K-nearest neighbors algorithm — Wikipedia, the free encyclopedia*, [Online; accessed 21-June-2021], 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=K-nearest_neighbors_algorithm&oldid=1029346235.

[33] *Support vector machine (svm)*. [Online]. Available: https://es.mathworks.com/discovery/support-vector-machine.html.

[34] S. Yıldırım, *Hyperparameter tuning for support vector machines - c and gamma parameters*, Jun. 2020. [Online]. Available: https://towardsdatascience.com/hyperparameter-tuning-for-support-vector-machines-c-and-gamma-parameters-6a5097416167.

[35] C. Busso *et al.*, "IEMOCAP: Interactive emotional dyadic motion capture database," *Language Resources and Evaluation*, vol. 42, no. 4, pp. 335–359, 2008. [Online]. Available: https://sail.usc.edu/iemocap/Busso_2008_iemocap.pdf.

[36] R. Lotfian and C. Busso, "Building naturalistic emotionally balanced speech corpus by retrieving emotional speech from existing podcast recordings," *IEEE Transactions on Affective Computing*, vol. PP, pp. 1–1, Aug. 2017. DOI: 10.1109/TAFFC.2017.2736999.

[37] F. Eyben, in *Real-time Speech and Music Classification by Large Audio Feature Space Extraction*. Springer International Publishing, 2016.

[38] T. Giannakopoulos, *Intro to audio analysis: Recognizing sounds using machine learning*, Accessed: 14-06-2021, Sep. 2020. [Online]. Available: https://medium.com/behavioral-signals-ai/intro-to-audio-analysis-recognizing-sounds-using-machine-learning-20fd646a0ec5.

[39] B. Logan, "Mel frequency cepstral coefficients for music modeling," *Proc. 1st Int. Symposium Music Information Retrieval*, Nov. 2000.

[40] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980. DOI: 10.1109/TASSP.1980.1163420.

[41] P. Calleja Acosta, "Verificación automática del locutor en sistema para el control de acceso," Jun. 2014.

[42] S. Li and A. Jain, *Encyclopedia of Biometrics*. Jan. 2009. DOI: 10.1007/978-0-387-73003-5.

[43] K. Werner and F. Germain, "Sinusoidal parameter estimation using quadratic interpolation around power-scaled magnitude spectrum peaks," *Applied Sciences*, vol. 6, p. 306, Oct. 2016. DOI: 10.3390/app6100306.

[44] C. Bouman, "Cluster: An unsupervised algorithm for modeling gaussian mixtures," 2014.

[45] F. Eyben *et al.*, "The geneva minimalistic acoustic parameter set (gemaps) for voice research and affective computing," *IEEE Transactions on Affective Computing*, vol. 7, no. 2, pp. 190–202, 2016. DOI: 10.1109/TAFFC.2015.2457417.

[46] T. Giannakopoulos, "Pyaudioanalysis: An open-source python library for audio signal analysis," *PLOS ONE*, vol. 10, e0144610, Dec. 2015. DOI: 10.1371/journal.pone.0144610.

[47] B. Schuller *et al.*, "The interspeech 2016 computational paralinguistics challenge: Deception, sincerity and native language," Sep. 2016, pp. 2001–2005. DOI: 10.21437/Interspeech.2016-129.

[48] Wikipedia contributors, *Glove (machine learning) — Wikipedia, the free encyclopedia*, [Online; accessed 8-June-2021], 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=GloVe_(machine_learning)&oldid=1026975560.

[49] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," vol. 14, Jan. 2014, pp. 1532–1543. DOI: 10.3115/v1/D14-1162.

[50] B. Tris Atmaja and M. Akagi, "Dimensional speech emotion recognition from speech features and word embeddings by using multitask learning," *APSIPA Transactions on Signal and Information Processing*, vol. 9, May 2020. DOI: 10.1017/ATSIP.2020.14.

[51] L. I.-K. Lin, "A concordance correlation coefficient to evaluate reproducibility," *Biometrics*, vol. 45, no. 1, pp. 255–268, 1989. [Online]. Available: http://www.jstor.org/stable/2532051.

[52] Wikipedia contributors, *Early stopping — Wikipedia, the free encyclopedia*, [Online; accessed 8-June-2021], 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Early_stopping&oldid=1006394815.

[53] A. Lee, *Choosing a baseline accuracy for a classification model*, May 2021. [Online]. Available: https://towardsdatascience.com/calculating-a-baseline-accuracy-for-a-classification-model-a4b342ceb88f.

[54] X. Mao, S. Chang, J. Shi, F. Li, and R. Shi, "Sentiment-aware word embedding for emotion classification," *Applied Sciences*, vol. 9, p. 1334, Mar. 2019. DOI: 10.3390/app9071334.

[55] *Portada del banco mundial*, Accessed: 06-05-2021. [Online]. Available: https://www.bancomundial.org/es/home.

[56] L. Sacristan, *España es el quinto pais de la ue con mejor conectividad: 80% de cobertura de fibra frente al 34% de media europea*, Accessed: 06-05-2021, Jun. 2020. [Online]. Available: https://www.xatakamovil.com/conectividad/espana-quinto-pais-ue-mejor-conectividad-80-cobertura-fibra-frente-al-34-media-europea.

[57] Vodafone, *Observatorio vodafone de la empresa 2019*, Accessed: 06-05-2021, 2019. [Online]. Available: https://xh4y28w4m30fiwf22ex7gvfa-wpengine.netdna-ssl.com/wp-content/uploads/2019/11/OVE_III-Estudio-sobre-el-Estado-de-la-Digitalizacio%CC%81n-Resumen-Ejecutivo.pdf.

[58] D. O. de la Union Europea, *Reglamento (ue) 2016/679 del parlamento europeo y del consejo de 27 de abril de 2016 relativo a la proteccion de las personas fisicas en lo que respecta al tratamiento de datos personales y a la libre circulacion de estos datos y por el que se deroga la directiva 95/46/ce (reglamento general de proteccion de datos)*, Accessed: 06-05-2021, Apr. 2016. [Online]. Available: https://www.boe.es/doue/2016/119/L00001-00088.pdf.

[59] *Lex access to european union law*, Accessed: 06-05-2021. [Online]. Available: https://eur-lex.europa.eu/legal-content/ES/ALL/?uri=CELEX%3A32016R0679.

[60] *Une-iso 21500:2013*, Accessed: 06-05-2021. [Online]. Available: https://www.aenor.com/normas-y-libros/buscador-de-normas/UNE?c=N0050883.

[61] *A guide to the project management body of knowledge: (PMBOK guide)*. Project Management Institute, 2017.

[62] Dec. 2018. [Online]. Available: https://www.boe.es/buscar/pdf/2018/BOE-A-2018-16673-consolidado.pdf.

[63] M. de Cultura, *Real decreto legislativo 1/1996, de 12 de abril, por el que se aprueba el texto refundido de la ley de propiedad intelectual, regularizando, aclarando y armonizando las disposiciones legales vigentes sobre la materia.* Apr. 1996. [Online]. Available: https:%20//www.boe.es/buscar/pdf/1996/BOE-A-1996-8930-consolidado.pdf.