Universitat Oberta de Catalunya (UOC)

Máster Universitario en Ciencia de Datos (*Data Science*)

TRABAJO FINAL DE MÁSTER

Área: Área 1

Distant galaxies analysis with Deep Neural Networks.

Autor: Raúl Cacho Martínez

Tutor: Anna Bosch Rue

Profesor: Xabier San Martín Aguirrezabal

Madrid, January 8, 2020

Créditos/Copyright

FICHA DEL TRABAJO FINAL

Título del trabajo:	Distant galaxies analysis with Deep Neural Networks.
Nombre del autor:	Raúl Cacho Martínez
Nombre del colaborador/a docente:	Anna Bosch Rué
Nombre del PRA:	Albert Solé Ribaltal
Fecha de entrega (mm/aaaa):	01/2020
Titulación o programa:	Máster Universitario en Ciencia de Datos
Área del Trabajo Final:	Área 1
Idioma del trabajo:	Inglés
Palabras clave	Deep-Learning, Galaxies, Stellar populations

Dedicatoria/Cita

A Claudia, Samuel y Amanda.

Agradecimientos

En primer lugar, quiero agradecer a Anna Bosch Rué, de la Universitat Oberta de Cataluña (UOC), su apoyo y su ayuda durante este trabajo. Siempre estuvo para resolver las dudas que tenía y para guiarme correctamente en el desarrollo de esta actividad. También quier darle las gracias a Albert Solé, también de la UOC, tanto por su ayuda como por sus consejos al elegir la temática del proyecto. Por último, quiero agradecer a los miembros del tribunal de evaluación el tiempo empleado en la lectura de la memoria y en la evaluación del conjunto del trabajo.

Es justo agradecer al equipo de la exploración ALHAMBRA, liderados por el Dr. Mariano Moles, gracias a los cuales son accesibles los datos del catálogo, que cuentan con una calidad excelente, y sin los cuales no habría sido posible este proyecto.

Por último, quiero agradecer a mi pareja, Claudia Rodríguez el apoyo que me ha prestado no sólo durante este proyecto, sino durante todo este Máster.

Abstract

In this work we face a very common problem in Astrophysics. One of the first parameters to obtain from a galaxy spectrum is the redshift. The redshift at which a galaxy is, can tell us a lot of things about the large scale structure of the Universe. However, the telescope time is limited, and it would take a lot of time to survey the whole sky observing the spectrum of galaxies. This is the reason why surveys using narrowband photometry (for example ALHAMBRA or JPAS) are arising. These surveys allow to observe a large number of galaxies in much less time than using spectroscopy, thus making astronomers able to disentangle the structure of the Universe and the features of very distant galaxies.

Traditionally, the features have been derived using the technique known as SED-fitting, which consists in deriving the features of the galaxy from its spectrum. This is not an easy problem, not only because of the large number of variables in play (velocity, velocity dispersion, age and metallicity for each single stellar population, or SSP), but because of the degeneracies. A degeneracy happens when two different SSPs show almost undistinguishable spectra. For example, a degeneracy exists between age and metallicity, with and old and metal-rich[1] SSPs showing similar spectrum to that of a young and metal-poor SSPs.

In this Master Thesis we evaluate the ability of Deep Neural Networks, using as input the observations of a galaxy, to obtain the parameters of the galaxy (redshift, mass and galaxy type).

[1] In astrophysics, all elements different from Hydrogen and Helium are called metals

Resumen

En este trabajo vamos a afrontar un problema habitual en Astrofísica. Uno de los primeros parámetros a medir en el espectro de una galaxia es el *redshift* o desplazamiento al rojo. El desplazamiento al rojo de una galaxia puede dar mucha información acerca de la estructura a gran escala del Universo. Sin embargo, el tiempo de telescopi oes limitado, y llevaría mucho tiempo observar todo el cielo obteniendo el espectro de las galaxias. Por ello están proliferando los catálogos de observaciones basados en photometría de banda estrecha (por ejemplo, ALHAMBRA o JPAS). Estos catálogos permiten observar un gran núnero de galaxias en mucho menos tiempo que usando espectroscopía, permitiendo a los astrónomos desentrañar la estructura del Universo a gran escala y pudiendo medir las características de las gaalxias más lejanas.

Tradicionalmente, las características de las galaxias se ha obtenido usando una técnica conocida como *SED-fitting* o ajuste espectral. Esta técnica consiste en ajustar un espectro a las observaciones fotométricas, permitiendo obtener las características de la galaxia. Este problema no es sencillo, no solo por la gran cantidad de variables involucradas, sino también por las degeneraciones existentes. Una degeneración ocurre cuando dos poblaciones estelares simples (SSP) tienen espectros prácticamente indistinguibles a pesar de que sus parámetros son completamente diferentes. Es ampliamente conocida, por ejemplo, las degeneraciones existentes entre la edad y la metaliciad, por la que una galaxia vieja y rica en metales [2] tiene un espectro muy parecido al de una galaxia joven pobre en metales.

En este trabajao evaluaremos la capacidad de Redes Neuronales Profundas de, usando como entrada las observaciones de una galaxia, obtener los parámetros fundamentales de dicha galaxia (desplazamiento al rojo, masa, y tipo de galaxia).

Palabras clave: Galaxies, stellar decomposition, inversion problem, stellar content, Deep Neural Networks, Deep Learning.

[2] En astrofísica se llaman metales a todos los elementos más allá del Helio

Contents

List of Figures

List of Tables

Acronyms

AGN :	Active Galactic Nuclei
ALHAMBRA :	Advanced Large, Homogeneous Area Medium Band Redshift Astronomical Survey
AM :	Amplitude Modulation
CCD :	Coupled Charge Device
CNN :	Convolutional Neural Network
CV :	Computer Vision
DA :	Data Augmentation
DEC :	Declination
DL :	Deep Learning
DNN :	Deep Neural Network
FM :	Frequency Modulation
FT :	Fourier Transform
FWHM :	Full Width at Half Maximum
GANN :	Generative Adversarial Neural Network
GTC :	Gran Telescopio Canarias
INN :	Invertible Neural network
JPAS :	Javalambre Physics of the Accelerating Universe Astrophysical Survey
ML :	Machine Learning
NaN :	Not a Number
NLP :	Natural Language Processing
NN :	Neural Network
PSK :	Phase Shift Keying
RA :	Right Ascension
S/N :	Signal to Noise Ratio
SDSS :	Sloan Digital Sky Survey
SED :	Spectral Energy Distribution

SNR :	Signal to Noise Ratio
SP :	Signal Processing
SSP :	Single Stellar Population
TMT :	Thirty Meter telescope
WT :	Wavelet Transform

Chapter 1

Introduction

1.1 Description

Spectral Synthesis is a well known and very extended technique to recover the stellar parameters of populations underlying the spectra (or the spectral energy distribution, SED) of galaxies, with a lot of tools already developed for this purpose (see, for example http://www.sedfitting.org/Fitting.html.

The most advanced codes (like STARLIGHT, Cid Fernandes et al. 2005; STECKMAP,Ocvirk et al. 2006a,b; or PPxF, Cappellari and Emsellem 2004) can recover non-parametric distributions of kinematics. However, it is still complicated to establish a clear relationship between age, metallicity and kinematics for each stellar population (see Chapter 4 in Cacho, 2015). Despite some efforts have been made in this direction, the results are still not accurate, as some degeneracies are involved in the process, and the uncertainties in the models are still large.

However, the high resolution SED fitting techniques can only be applied when nearby galaxies are involved. For distant galaxies, SED fitting must be based on narrow band filters, seeming like low resolution spectroscopy. In the last few years, some projects like ALHAMBRA or JPAS have succeeded in observing galaxies with a large number of overlapping filters.

The main advantage of this technique is that imaging is much less time-consuming than spectroscopy, and large areas of the sky can be observed in a single shot. Therefore, a larger number of galaxies can be observed, highly increasing the amount of data available for astrophysical studies.

1.2 Motivation

There are several motivations to achieve the goals identified in Sect. 1.3. The main motivation is continuing the work started during my PhD (Cacho, 2015). One of the main caveats was the lack of consistency in the results obtained with different codes devoted to recovering the stellar properties of galaxies. This was specially true when more then two SSPs were present in the spectrum. Moreover, in Astrophysics, the implantation of Machine/Deep Learning is not yet fully implemented. Therefore, there are not many tools using ML/DL. In fact most of them are devoted to Computer Vision to identify, or classify different features in celestial bodies (see, for example Tuccillo et al., 2016; Hon et al., 2018; Silburt et al., 2019; Gillet et al., 2019; Herbel et al., 2018).

A secondary motivation is my present job, in which we want to initiate Machine Learning into some projects, in order to detect, identify and classify different types of electromagnetic emissions. Therefore, getting some expertise in the use of deep neural networks (DNN) in the analysis of electromagnetic signals could be of great advantage.

1.3 Goals

During the implementation of this project, it came out that the original goals could not be achieved within the scope of a Master Thesis. Therefore, the goals had to be changed. The original goals will be described in Appendix B.1. The final objectives of this work are the following:

1. Learn the whole life cycle of data, in particular the capture, the cleaning, the analysis and the visualization.

2. Being able to design one or several models capable of predicting different variables related to distant galaxies (redshift and stellar mass, among others).

3. To design, train and test a neural network, capable of accepting the emission of the galaxies in different predefined filters as the input, and returning a reliable array as close as possible to the parameters previously obtained for the galaxy using other techniques.

1.4 Methodology

To achieve the goals described in Sect 1.3, we will try to adapt some different algorithms used for a different purpose. For example, in telecommunications, Neural Networks are used to detect, identify and classify communication signals (O'Shea et al., 2016; West and O'Shea, 2017; Chen et al., 2019). Whilst the objectives are different, the fundamentals are similar, as there are a number of different signals underlying the spectrum. Moreover, these signals can be associated to templates modified under different conditions which can be parametrized.

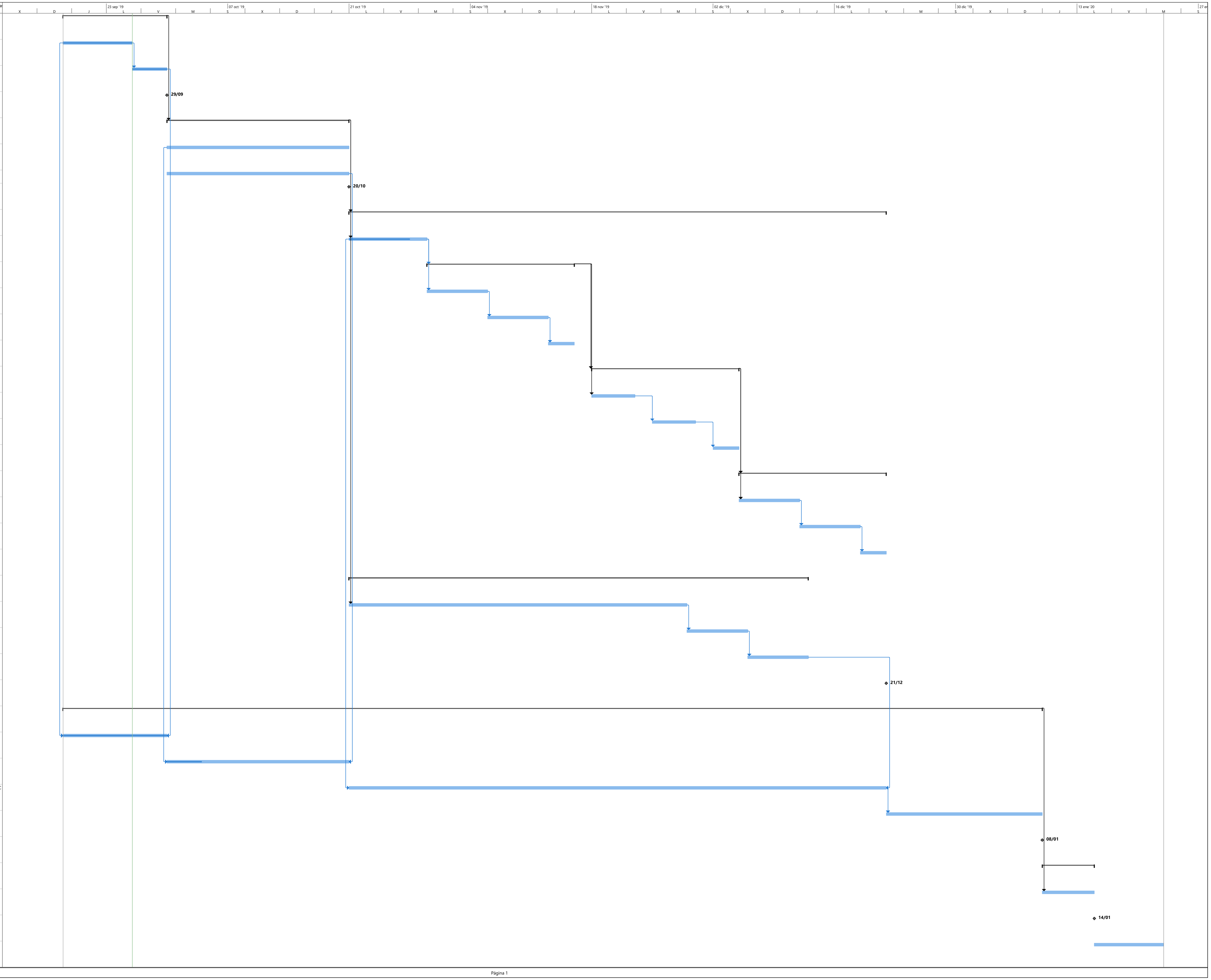
The tools to be used in the project are the following:

- Python 3.6.6, with the following frameworks and modules installed
  - AstroPY, version 3.2.1
  - ipython, version 6.5.0
  - keras, version 2.2.4 using TensorFlow backend
  - matplotlib, version 2.2.2
  - numpy, version 1.15.2
  - PyAstronomy, version 0.13.0
  - scipy, version 1.1.0
  - spectres
  - TensorFlow, version 1.12.0
- ProjectLibre and MSProject, for the planning
- SublimeText, for coding
- TeXworks, using MikTeX backend for writing the report

1.5 Planning

The next page shows a Gantt diagram in which high level tasks have been identified and planned. The beginning and end of the project correspond to the beginning and end of the semester. The Master Thesis was planned to span for 18 weeks. To cover the 12 ECTS credits, which correspond, approximately to 300h, the mean dedication should be around 17 hours a week.

Id	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1	**Definición y planificación del trabajo**	**9 días**	**mié 18/09/19**	**dom 29/09/19**	
2	Lectura de bibliografía	6 días	mié 18/09/19	mié 25/09/19	
3	Definición del trabajo	3 días	jue 26/09/19	dom 29/09/19	2
4	PEC 1	1 día	dom 29/09/19	dom 29/09/19	
5	**Estado del Arte**	**16 días?**	**lun 30/09/19**	**dom 20/10/19**	**1**
6	Análisis	16 días?	lun 30/09/19	dom 20/10/19	
7	Búsqueda de información bibliográfica y consulta a expertos	16 días	lun 30/09/19	dom 20/10/19	
8	PEC 2	1 día	dom 20/10/19	dom 20/10/19	
9	**Diseño e Implementación del Trabajo**	**47 días**	**lun 21/10/19**	**sáb 21/12/19**	**5**
10	Creación de Conjunto de Datos	8 días	lun 21/10/19	mar 29/10/19	5
11	**1er ciclo**	**13 días**	**mié 30/10/19**	**vie 15/11/19**	**10**
12	Diseño de la red	5 días	mié 30/10/19	mar 05/11/19	10
13	Entrenamiento	5 días	mié 06/11/19	mar 12/11/19	12
14	Pruebas	3 días	mié 13/11/19	vie 15/11/19	13
15	**2o ciclo**	**13 días**	**lun 18/11/19**	**mié 04/12/19**	**11**
16	Diseño de la red	5 días	lun 18/11/19	vie 22/11/19	11
17	Entrenamiento	5 días	lun 25/11/19	vie 29/11/19	16
18	Pruebas	3 días	lun 02/12/19	mié 04/12/19	17
19	**3er ciclo**	**13 días**	**jue 05/12/19**	**sáb 21/12/19**	**15**
20	Diseño de la red	5 días	jue 05/12/19	mié 11/12/19	15
21	Entrenamiento	5 días	jue 12/12/19	mié 18/12/19	20
22	Pruebas	3 días	jue 19/12/19	sáb 21/12/19	21
23	**Comparación con otras herramientas**	**40 días**	**lun 21/10/19**	**jue 12/12/19**	
24	Ejecución de otro SW	30 días	lun 21/10/19	jue 28/11/19	5
25	Definición de Pruebas de comparaci	5 días	vie 29/11/19	jue 05/12/19	24
26	Comparación de resultados	5 días	vie 06/12/19	jue 12/12/19	25
27	PEC 3	1 día	sáb 21/12/19	sáb 21/12/19	
28	**Redacción de la memoria**	**86 días**	**mié 18/09/19**	**mié 08/01/20**	
29	Primer borrador	9 días	mié 18/09/19	dom 29/09/19	2CC;3FF
30	Introducción	16 días	lun 30/09/19	dom 20/10/19	6CC;7FF
31	Cuerpo de la memoria	47 días	lun 21/10/19	sáb 21/12/19	26FF;10CC
32	Conclusiones	14 días	dom 22/12/19	mié 08/01/20	31
33	PEC 4	1 día	mié 08/01/20	mié 08/01/20	
34	**Presentación y defensa del proyecto**	**4 días**	**jue 09/01/20**	**mar 14/01/20**	
35	Preparación de la presentación	4 días	jue 09/01/20	mar 14/01/20	28
36	PEC 5	1 día	mar 14/01/20	mar 14/01/20	
37	Defensa Pública	6 días	mié 15/01/20	mié 22/01/20	



Chapter 2

State of the Art

2.1 Computer Vision and Signal Processing

Computer vision (CV) and feature detection are among the more prominent techniques when talking about Machine Learning. A quick look at https://paperswithcode.com/sota shows that there are around 700 papers related to CV, whilst the next discipline, Natural Language Processing (NLP) has approximately 1/3 of publications.

Very similar to CV is signal processing (SP). Both of them try to find and recognize features to identify objects in pictures or signals in electromagnetic spectra. Li et al. (2019) reviews algorithms and different approaches to signal analysis, in particular, to modulation recognition. It has to be mentioned that these techniques work under a series of premises:

- They are used on communication signals, which have a dominant periodic component (however, some non-periodic signals can be found along with the carrier periodic wave). This opens the door to a simplification of the problem using analytical techniques, such as Fourier or Wavelet Transforms (FT, WT)

- The modulation and features of the signals are finite, and well known. You only have to describe the frequency of the signal (probably not known, but constrained by the brand and model of the radio-frequency receivers) and the modulation[1] (which can be analog: AM, FM, ...; or digital: ASK, APSK, QPSK, 8PSK,...). Once these two parameters are known, the signal can be demodulated, and its content accessed (unless it is encrypted).

- The signals do not overlap, this is, there is only a signal at a given frequency. In scenarios in which two or more signals of the same frequency arrive to the receiver simultaneously, the performance of these algorithms drops dramatically.

[1] In https://en.wikipedia.org/wiki/Modulation further explanation and a list of possible modulations can be found.

2.2 Machine Learning in Astrophysics

Machine and Deep learning are just incipient techniques in Astrophysics. They are mostly used for data mining, and for the analysis of large amount of data, understanding this as lots of objects (galaxies, stars, etc.). In fact, the most extended use is for classifying images in large surveys. For example, Convolutional Neural Networks have been used to discriminate between stars and galaxies (Kim and Brunner, 2016) in the Sloan Digital Sky Survey (SDSS) or for classification galaxies by their morphology (Katebi et al., 2019; Walmsley et al., 2019) in the Galaxy Zoo Project [2].

Another extended use of Machine Learning in Astrophysics is what is known as Data Augmentation (DA). DA consists in taking a dataset and, by transforming it in permitted ways (for example, an image can be stretched, flipped, cropped, ...), increasing the number of different elements in the dataset. This can be useful to generate synthetic galaxies, in order to simulate surveys, or how galaxies with given features will be observed through telescopes. (Ma et al., 2018; Regier et al., 2018; Fussell and Moews, 2019; Reiman and Göhre, 2019; Khan et al., 2019).

2.3 Stellar Population Synthesis

Despite Stellar Population Synthesis is one of the most powerful techniques to understand the behavior and evolution of galaxies, not many steps forward have been achieved. For example, one of the latest code released, `FIREFLY` (Wilkinson et al., 2017), still makes use of "classical" statistics to get the results.

However, there are some groups trying to introduce Machine Learning for this purposes. For example Salvato et al. (2019) approaches the inversion problem to SED[3] fitting. Also Simet et al. (2019); Lovell et al. (2019) use Machine Learning to derive galactic properties from the spectrum of galaxies. However, they only take into account the Star Formation History of galaxies. Kinematics is not considered, and it is well known that degeneracies exist between kinematics and age, leading to wrong results if kinematics is not taken into account.

2.4 Narrow band filters surveys

During the late 80's, the first attempt to survey the whole sky, the Digital Sky Survey (DSS), was made. It consisted of a compilation of photographic plates taken from telescopes around the world. It was taken in two different epochs, using three different filters (red, blue and

[2] also based on SDSS

[3] SED comes from Spectral Energy Distribution, and usually corresponds to low resolution spectra of galaxies

infrared).

Almost two decades later, in 2000, a pilot study was launched to observe the sky in different filters. In particular, 5 different filters were used(called u, g, r, i, z), from ultraviolet (u) to infrared (z). Not only photometric data was taken, but also, using optical fibers, the spectra of the brightest object was taken, allowing for direct comparison between photometric and spectroscopic data, proving the capability of measuring galactic parameters from images instead of spectra.

This studies led to the appearance of different surveys[4] using narrow band filters to observe and study distant galaxies. The data of two of the most important surveys (ALHAMBRA and JPAS) was taken in Spanish observatories.

2.4.1 ALHAMBRA survey

ALHAMBRA (Advanced Large, Homogeneous Area Medium Band Redshift Astronomical Survey) is a survey (and a catalog with the same name) of galaxies taken from the 3.5m telescope in Calar Alto (Almería, Spain). This survey observed well known fields of galaxies (previously observed with other space and ground-based telescopes, like Hubble, or Spitzer), in order to measure the redshift of the galaxies in the fields, among other magnitudes.

The fields were observed using 24 different narrow band filters (see Fig. 2.1.

Finally, the catalog consists in the flux of around 450,000 objects in the different filters in which they were observed (see section 3.1 for details).

2.4.2 JPAS survey

J-PAS (Javalambre Physics of the Accelerating Universe Astrophysical Survey[5]) will cover at least 8,000 deg^2 in approximately 5 years, using an unprecedented system of 56 narrow band filters in the optical. The filter system (see Fig. 2.2) was optimized to pursue three main scientific goals: first, to accurately measure photometric redshifts for galaxies; second, to study stellar populations in nearby galaxies; and third, to resolve broad spectral features of objects such as AGNs and supernovae.

It is being performed in the Observatorio Astronómico de Javalambre, in Teruel (Spain), and plans to observe around half a billion of galaxies. This survey is still in progress, with the first data released in December 2019, covering 1 deg^2, and with data of more than 64,000 galaxies

[4] http://alhambrasurvey.com/otros_surveys.php

[5] http://www.j-pas.org/

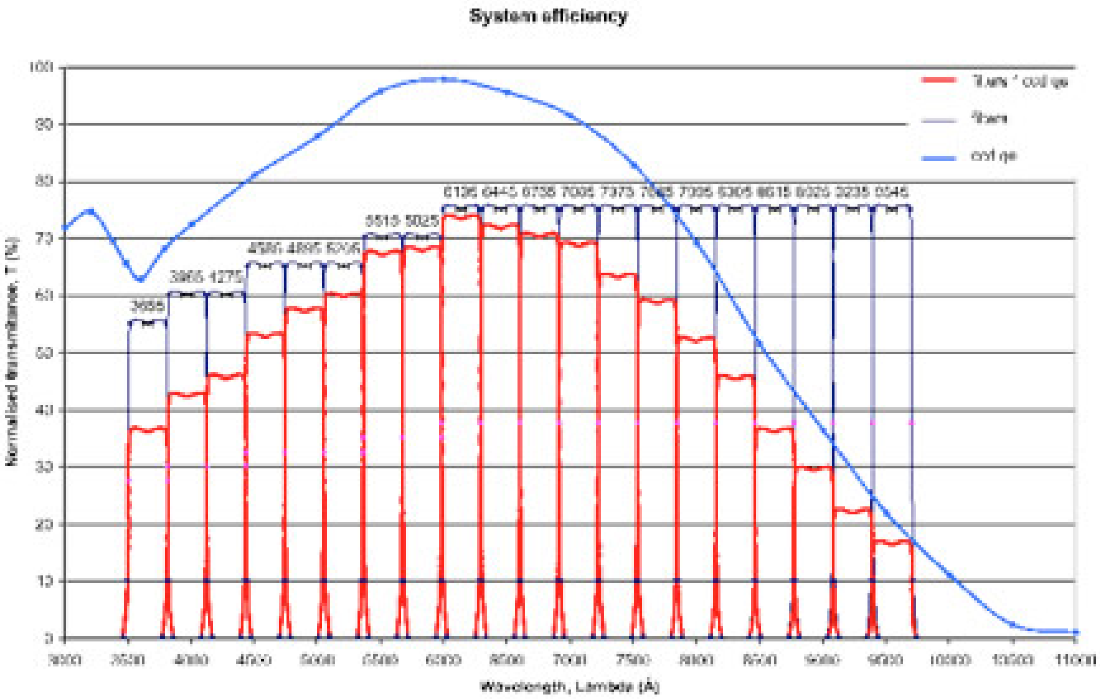


Figure 2.1: Wavelength coverage and spectral transmission of the filters used in ALHAMBRA Survey

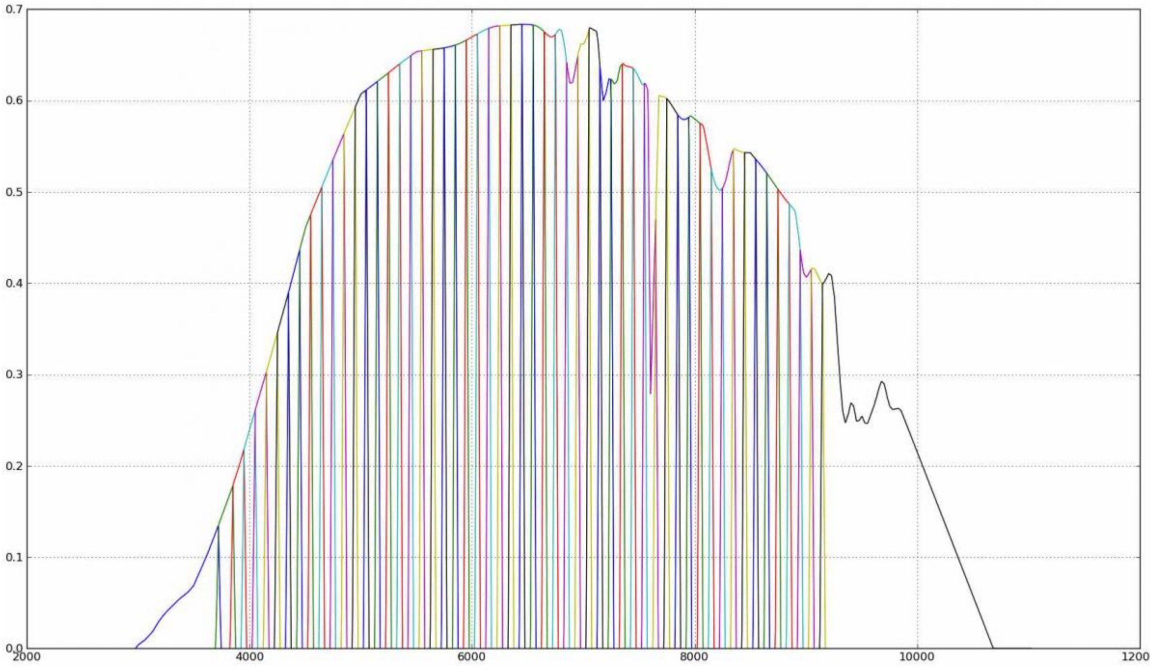


Figure 2.2: Wavelength coverage and spectral transmission of the filters used in JPAS Survey

Chapter 3

Implementation

3.1 The data

The data used for this Master Thesis is the result of the ALHAMBRA survey. ALHAMBRA is a photometric survey devoted to the observation of distant galaxies in some well known fields. The main goal of the survey is to measure the flux of galaxies in different narrow band filters (each filter overlapping with the adjacent ones) in order to estimate the *stellarity*[1], the range of minimum and maximum possible redshift, along with the most probable one, and the mass of stars in the galaxy, derived from its luminosity.

The data is publicly available, and can be downloaded from the web of the project [2]. The data consists of a csv file with the data of 446,343 objects. For each object, among other, the following information can be found:

- ID (int): Identification number of the observed object.
- ObjID (int): Alhambra Identification of the detection, constructed using the detection image (3 digits), the field (1 digit), the pointing (2 digits), the CCD (1 digit) and the Filter set ID (5 digits).
- Field (int): The field where the object is in.
- Pointing (int): The pointing of the telescope within the field.
- CCD (int): the number of the sensor in which the object was captured.
- RAdeg and DECdeg (float): the coordinates of the sky in which the object can be found.

[1] The probability that a given object is a star or a galaxy

[2] http://svo2.cab.inta-csic.es/vocats/alhambra/download/alhambra.csv.gz

- x, y (float): coordinates on the CCD of the centroid of the object.
- area, fwhm (float): area (in pixels and arcsec) covered by the 90% of the light of the object.
- stell (float): indicator of the object being a star or a galaxy.
- ell (float): ellipticity of the object.
- a, b, theta (float): semi major, semi minor axis and orientation of the major axis of the object.
- RK, RF (float): Kron apertures and fraction-of-light radii.
- FXXXW, dFXXXW (float): Flux and associated uncertainty of the object on the filter with central wavelength XXX.
- F814W_3arcs, dF814W_3arcs (float): Flux and associated uncertainty contained in an aperture of diameter 3 arcsec in the image obtained with the filter F814.
- zb_1, zb_min_1, zb_max_1 (float): most probable redshift, and minimum and maximum possible values of the redshift for the observed object.
- tb_1 (int): spectral type of the object.
- Stellar_Mass (float): Mass of the stellar content of the galaxy.
- M_ABS_1: absolute (AB) magnitude for the Johnson B-band.

3.2 Preprocessing

Data is read as a pandas dataframe. Most of the magnitudes of interest are well determined, and there is no need for interpolating or removing NaNs. However, sometimes the flux in some filter may be too low or too high, being marked by 99 and -99, respectively. These are very extreme values, they were clipped to 0 and 27. These values are not randomly defined, as there will be no galaxies with fluxes over or under them.

For the categorical values, we convert the values to categorical data, considering the ranges in which each category falls.

The need of normalizing the data was also explored. A comparison was made between the results obtained using normalized and not-normalized data, obtaining similar results (with the obvious exception of the value range). Therefore we decided to not normalize the data.

3.3 Modeling

The main goal is to predict the *stellarity*, redshift, spectral type, stellar mass of a galaxy, and absolute magnitude of a galaxy using the observed flux in each of the filters.

One possible solution is to implement a DNN which would take as the input the measured values of the flux in each filter. However, as the predicted variables have different natures, we could not use only one single neural network for the task: stellarity is a categorical variable (with 2 categories), whilst the other parameters are continuous magnitudes.

Moreover, a large amount of cross talk among the outputs was detected. Therefore the best strategy is to build four independent Neural Networks, one for each predicted variable, and ensemble them to work as one neural network instead of four independent networks.

3.3.1 Neural network for redshift

This is a neural network designed as a correlation model between the input and the predicted redshift. The neural network consists in an input layer of 75 elements and a hidden layer of 120 elements. These two layers have a linear activation function. The output layer consists of 1 neuron with no activation function.

Adam is the chosen optimizer, "mean square error" the loss function and r^2 the metric. Please note that the regression is not linear, and therefore r^2 is not a valid measurement of the goodness of the fit, but only an indicator of it. This can be seen as r^2 cannot be higher than 1 in linear regression.

No overfitting was detected, so no regularization is needed.

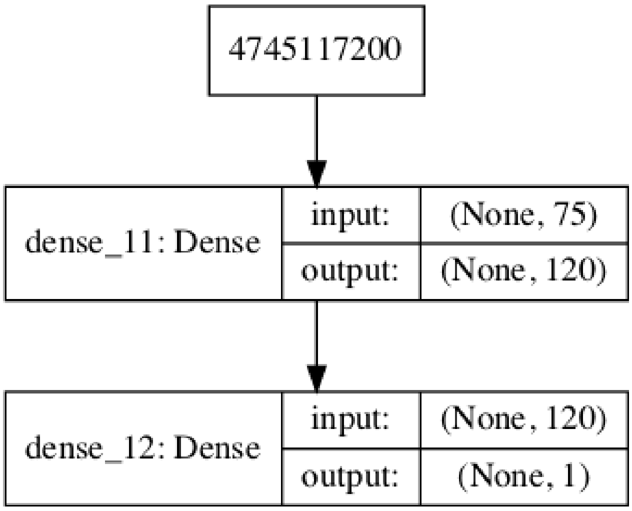


Figure 3.1: Model designed for the calculation of redshift.

3.3.2 Neural network for stellarity

In this case the neural network is built for classification. An architecture with 4 layers has been implemented: the input layer has 75 neurons with a *relu* activation function; the output layer has two neurons (one for each category) and *softmax* activation function; and two hidden layers were added with 500 and 600 neurons respectively, with *relu* activation functions.

In this case, the optimizer is *adam*, the loss function is "categorical crossentropy" and the metric is "accuracy".

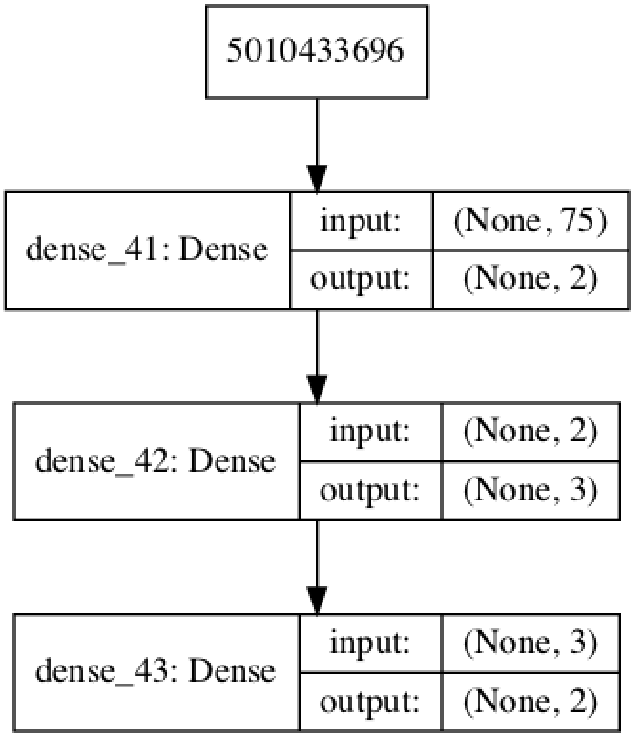


Figure 3.2: Model designed for the calculation of stellarity.

3.3.3 Neural Network for Stellar Mass

This network is similar to that for the redshift, as this network is also a regression model. In this case, there are three hidden layers with 120, 200 and 50 elements with linear activation function (as shown in Fig. 3.3). For this neural network we use *adam* optimizer, with *mse* and r^2 as loss function and metrics, respectively.

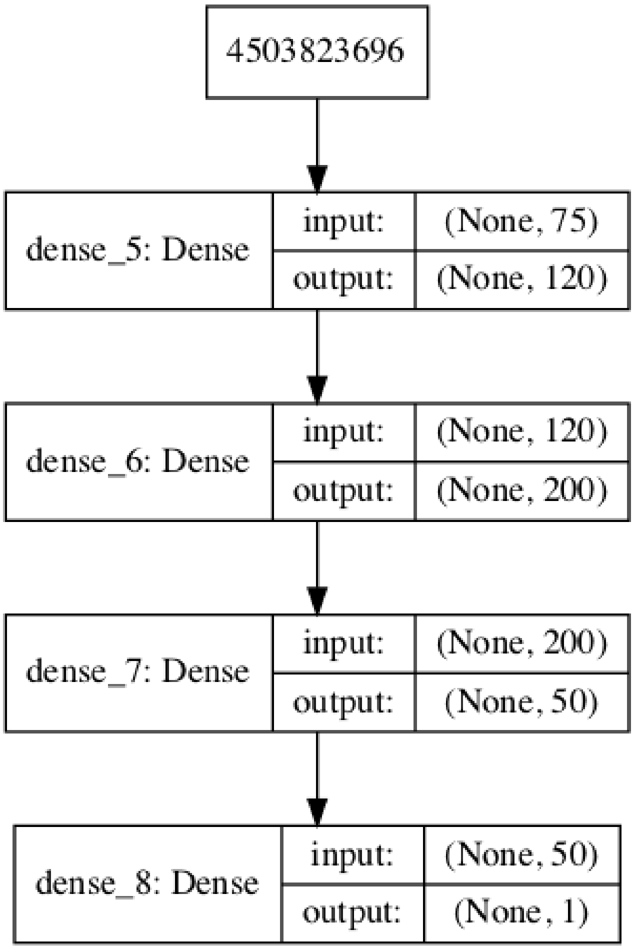


Figure 3.3: Model designed for the calculation of Stellar Mass.

3.3.4 Neural network for Spectral Type

This network is devoted to regression, as the spectral type is a continuous magnitude. This network consists in an input layer of 75 elements, an output layer of 5 elements (as there are 5 categories) and 3 hidden layers with 120 and 500 neurons.

3.3.5 Training

First of all, the optimal training epochs for these neural networks has to be decided. A training for a large number of epochs is performed, plotting both loss function and metrics. The optimal number of epochs is, approximately, the number when both the loss and metrics functions start to settle. To do this, one random sample of 10% (44,635) elements from the original sample was built for training and a sample of 1% (4,463) of the galaxies for validating.

The optimal number of epochs is as follows:

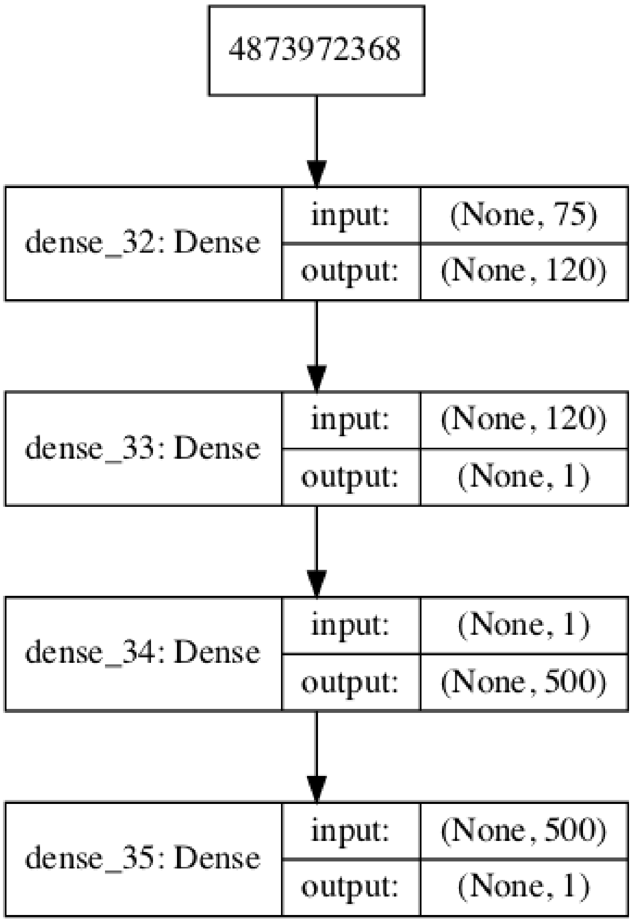


Figure 3.4: Model designed for the calculation of Spectral Type.

- redshift: 80 epochs
- stellarity: 20 epochs
- stellar mass: 7 epochs
- spectral type: 10 epochs

Once the optimal number of epochs is established, the neural networks are built using a 95% of the sample (424,025 objects) for training, and a 5% (22,317 objects) for validation.

3.3.6 Postprocessing

Once the results are obtained, they must be turned into valid values with physical sense. Therefore the predicted data has to be transformed, using the inverse transformations in the opposite order as we did for the output data used for training. To undo the normalization, the results have to be multiplied by the range of the original variable and add the minimum value of that variable in the dataset, so the original range of the variable can be recovered.

As it will be shown in Sect. 4, the correlation between expected and predicted values for the neural networks devoted to regression is strong, but there is a need for a correction to recover the original values. This correction consists in fitting a high order polynomial between the expected and the predicted values. Once the value for a particular input is predicted, we apply the polynomial on that value to get the final result.

3.3.7 Ensemble

Instead of combining the neural networks and combine the results, the best option was to handle each result in a separate way. However, a short function was written to simplify the calculation of the results from a single input, returning an array with all the outputs from the neural networks.

Chapter 4

Results

4.1 Redshift

Figure 4.1 shows a comparison of the predicted (output) and the expected results for the redshift. As it can be seen there is a linear positive correlation between both of them below $z = 0.9$. However, over this value, the results cannot be trusted, as there is more than one possible value of the expected redshift for each predicted redshift.

This issue responds to the fact that there are much less galaxies with redshifts between 0.9 and 1.0, as it will be discussed later.

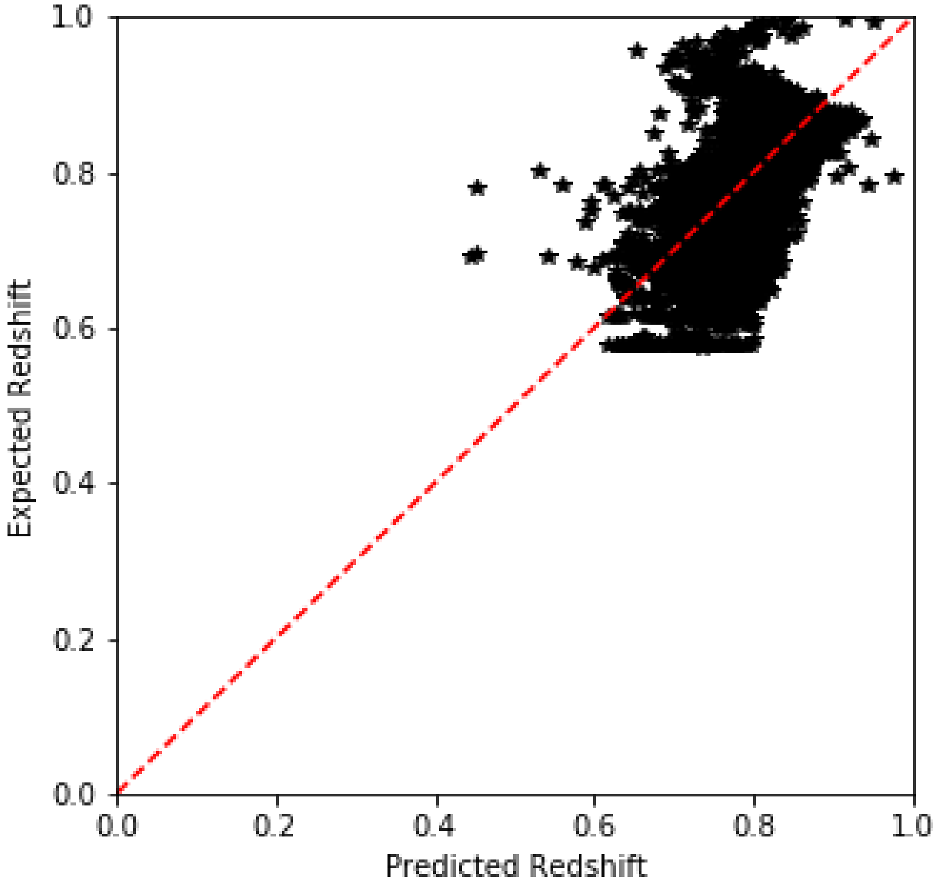


Figure 4.1: Comparison of the prediction as a function of the expected output for the redshift.

4.2 Stellarity

To analyze the accuracy of the calculation of the parameter compare the predicted and expected labels is compared (see Table 4.1). The table shows the predicted and expected labels for the test set. The table shows that, for the 22318 items in the set, 22106 were classified correctly, whilst 212 were misclassified. This means around a 99% accuracy in the prediction. Figure 4.2 shows the confusion matrix obtained for the stellarity.

		Expected	
		Galaxy	Star
Predicted	Galaxy	21700	23
	Star	189	406

Table 4.1: Predicted and expected frequencies of the stellarity

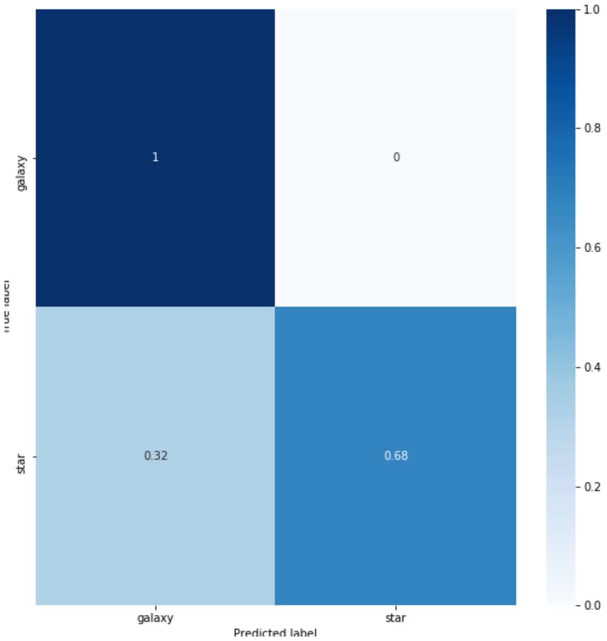


Figure 4.2: Confusion matrix for the parameter *stellarity*.

4.3 Stellar Mass

Figure 4.3 shows the predicted stellar mass against the expected mass for the test dataset. In this case, there is a linear positive correlation between both magnitudes. However, the slope of the correlation is not the unity.

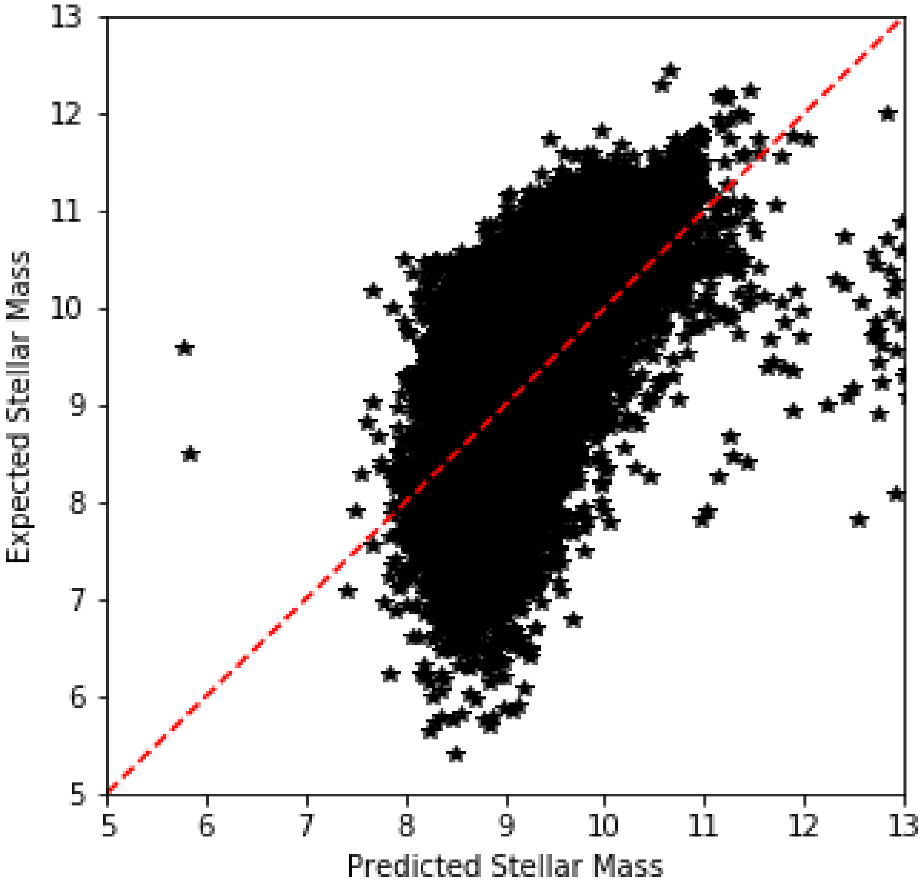


Figure 4.3: Comparison of the values for the parameter *Stellar Mass*.

4.4 Spectral Type

Figure 4.4 shows the predicted value for the galaxies in the test dataset, as a function of the expected value. In this occasion, the correlation is clear, with a correlation between them close to one.

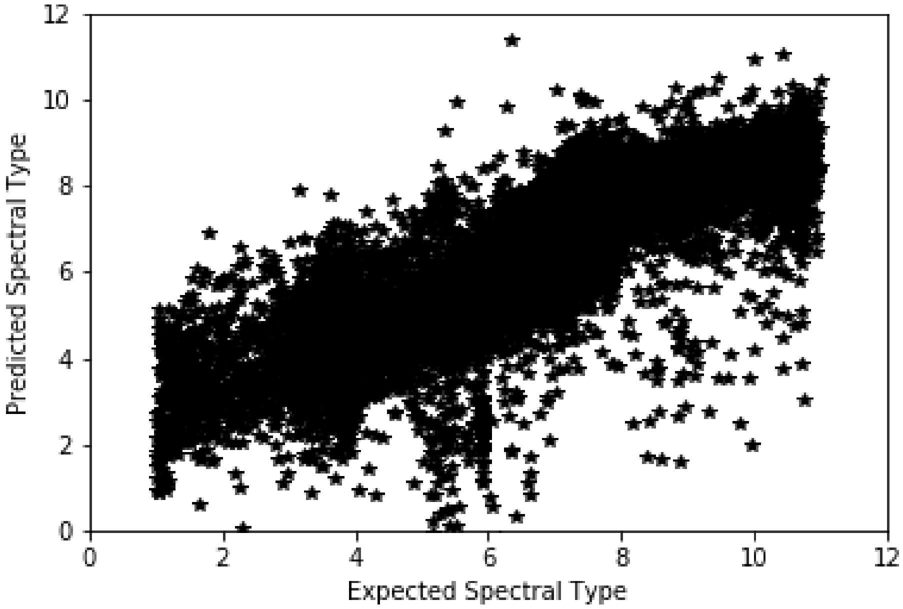


Figure 4.4: Comparison of the *Spectral type* for the test dataset.

Chapter 5

Summary and Conclusions

In this work it has been proven the capability of neural networks to solve a complex problem in Astrophysics as is the measurement of different properties in distant galaxies using only narrow band imaging of the galaxies. To achieve this main goal, the data had to be taken from the original source, "cleaned" and transformed prior to its analysis.

In total, four different models were developed in order to interpret the data and to predict the output (redshift, stellarity, stellar mass and spectral type) of new inputs consisting of the data taken from the observations.

We show the comparison between the estimated redshift using neural networks and the redshift obtained in ALHAMBRA survey. Whilst in ALHAMBRA the uncertainties are around $\frac{\Delta z}{1+z} = 0.03$, we obtain uncertainties lower than 0.006.

It is also shown how the neural network classifies the objects as stars or galaxies with an accuracy very close to 100%, with the advantage that no threshold or assumption has to be taken into account. Moreover, the algorithm considers not only the shape and the properties of an object in a single image, but also the observed flux in each of the filters.

Robust predictions of the stellar mass of a galaxy using neural networks can be performed. The mass is a magnitude which is tightly linked to the evolution of the Universe, and knowing the mass of the galaxies at different epochs, may shed some light on how the Universe evolved in its early stages.

Related to this is the spectral type of the galaxies. The spectral type gives an idea about how the galaxy is, and how it has evolved. Again, knowing the evolution of individual galaxies is really important to understand the evolution of the Universe itself.

With this proof of concept, we think that it is possible to use DL techniques to the study of distant galaxies, improving the telescope efficiency (as only images are needed, instead of spectroscopy, much more time consuming) and thus being able to observe more galaxies in the same amount of time.

With the advent of large telescopes (like GTC, TMT, etc...) the smaller telescopes will have a second youth as survey telescopes. Using DL in this surveys, can lead to increased efficiency and homogeneity, which may be a key for the interoperability of the small telescopes.

Chapter 6

Future Work

From the conclusions obtained in this work, the following lines are proposed for an improvement of the results and/or their application on different fields.

- The problem faced in this work is clearly a non-linear problem. Neural networks are non-linear regressors, so they should be able to solve non-linear regression problems. As it has been shown in Sect. 4, there is a margin for improvement. It is worth trying a different number of hidden layers with different numbers of neurons in each layer.

- The first dataset of JPAS survey has become publicly available recently. JPAS relies on the same concept as ALHAMBRA, this is, imaging galaxies in narrow band filters and calculating their parameters from the observation in those filters. The biggest difference is that JPAS uses 56 different filters, while ALHAMBRA consists of 20 filters. It would be interesting to test the capability of similar models to those developed in this work to predict the data in JPAS survey.

- Classical SED fitting methods for the estimation of photometric redshifts usually try to recover the spectrum of the galaxy. This is usually done by considering a reduced number of stellar populations (usually two), convolving the spectra with the transmission of each filter and comparing with the observed fluxes. It could be interesting to use DL techniques to estimate the spectrum. For example, a number of galaxy spectra could be taken from a library, and implement a Generative Adversarial Neural Network to predict the spectrum responsible of the measured emission in each of the narrow band filters.

- One extra step is to decrease the width of the filters. In the limit, there will be a convergence between narrow band photometry and low resolution spectroscopy. It could be studied if the techniques used in this work would be also valid when talking about spectroscopy. It would be necessary to estimate the minimum or maximum spectral resolution at which these techniques are still useful, as well as other parameter which are not taken into account here, like signal-to-noise ratio, intrinsic properties of galaxies, conditions in which the data were obtained, etc.

Bibliography

Cacho, R. (2015). Formation and evolution of galactic bulges: the importance of secular processes. Tesis Doctoral.

Cappellari, M. and Emsellem, E. (2004). Parametric Recovery of Line-of-Sight Velocity Distributions from Absorption-Line Spectra of Galaxies via Penalized Likelihood. *Publications of the Astronomical Society of the Pacific*, 116(816):138–147.

Chen, S., Zheng, S., Yang, L., and Yang, X. (2019). Deep learning for large-scale real-world ACARS and ADS-B radio signal classification. *CoRR*, abs/1904.09425.

Cid Fernandes, R., Mateus, A., Sodré, L., Stasińska, G., and Gomes, J. M. (2005). Semi-empirical analysis of Sloan Digital Sky Survey galaxies - I. Spectral synthesis method. *Monthly Notices of the Royal Astronomical Society*, 358:363–378.

Falcón-Barroso, J., Sánchez-Blázquez, P., Vazdekis, A., Ricciardelli, E., Cardiel, N., Cenarro, A. J., Gorgas, J., and Peletier, R. F. (2011). An updated MILES stellar library and stellar population models. *Astronomy & Astrophysics*, 532:A95.

Fussell, L. and Moews, B. (2019). Forging new worlds: high-resolution synthetic galaxies with chained generative adversarial networks. *Monthly Notices of the Royal Astronomical Society*, 485(3):3203–3214.

Gillet, N., Mesinger, A., Greig, B., Liu, A., and Ucci, G. (2019). Deep learning from 21-cm tomography of the cosmic dawn and reionization. *Monthly Notices of the Royal Astronomical Society*.

Herbel, J., Kacprzak, T., Amara, A., Refregier, A., and Lucchi, A. (2018). Fast point spread function modeling with deep learning. *Journal of Cosmology and Astroparticle Physics*, 2018(07):054–054.

Hon, M., Stello, D., and Zinn, J. C. (2018). Detecting solar-like oscillations in red giants with deep learning. *The Astrophysical Journal*, 859(1):64.

Katebi, R., Zhou, Y., Chornock, R., and Bunescu, R. (2019). Galaxy morphology prediction using capsule networks. *Monthly Notices of the Royal Astronomical Society*, 486(2):1539–1547.

Khan, A., Huerta, E., Wang, S., Gruendl, R., Jennings, E., and Zheng, H. (2019). Deep learning at scale for the construction of galaxy catalogs in the dark energy survey. *Physics Letters B*, 795:248–258.

Kim, E. J. and Brunner, R. J. (2016). Star–galaxy classification using deep convolutional neural networks. *Monthly Notices of the Royal Astronomical Society*, 464(4):4463–4475.

Li, X., Dong, F., Zhang, S., and Guo, W. (2019). A survey on deep learning techniques in wireless signal recognition. *Wireless Communications and Mobile Computing*, 2019.

Lovell, C. C., Acquaviva, V., Thomas, P. A., Iyer, K. G., Gawiser, E., and Wilkins, S. M. (2019). Learning the relationship between galaxies spectra and their star formation histories using convolutional neural networks and cosmological simulations.

Ma, Z., Zhu, J., Li, W., and Xu, H. (2018). Radio Galaxy Morphology Generation Using DNN Autoencoder and Gaussian Mixture Models. *arXiv e-prints*, page arXiv:1806.00398.

Ocvirk, P., Pichon, C., Lançon, A., and Thiébaut, E. (2006a). STEllar Content and Kinematics from high resolution galactic spectra via Maximum A Posteriori. *Monthly Notices of the Royal Astronomical Society*, 365:74–84.

Ocvirk, P., Pichon, C., Lançon, A., and Thiébaut, E. (2006b). STEllar Content from high resolution galactic spectra via Maximum A Posteriori. *Monthly Notices of the Royal Astronomical Society*, 365:46–73.

O'Shea, T. J., Corgan, J., and Clancy, T. C. (2016). Convolutional Radio Modulation Recognition Networks. In *Engineering Applications of Neural Networks*, pages 213–226. Springer International Publishing.

Regier, J., Miller, A. C., Schlegel, D., Adams, R. P., McAuliffe, J. D., and Prabhat (2018). Approximate Inference for Constructing Astronomical Catalogs from Images. *arXiv e-prints*, page arXiv:1803.00113.

Reiman, D. M. and Göhre, B. E. (2019). Deblending galaxy superpositions with branched generative adversarial networks. *Monthly Notices of the Royal Astronomical Society*, 485(2):2617–2627.

Salvato, M., Ilbert, O., and Hoyle, B. (2019). The many flavours of photometric redshifts. *Nature Astronomy*, 3(3):212–222.

Silburt, A., Ali-Dib, M., Zhu, C., Jackson, A., Valencia, D., Kissin, Y., Tamayo, D., and Menou, K. (2019). Lunar crater identification via deep learning. *Icarus*, 317:27–38.

Simet, M., Soltani, N. C., Lu, Y., and Mobasher, B. (2019). Comparison of observed galaxy properties with semianalytic model predictions using machine learning.

Tuccillo, D., Huertas-Company, M., Decencière, E., and Velasco-Forero, S. (2016). Deep learning for studies of galaxy morphology. *Proceedings of the International Astronomical Union*, 12(S325):191–196.

Walmsley, M., Smith, L., Lintott, C. J., Gal, Y., Bamford, S., Dickinson, H. J., Fortson, L., Kruk, S. J., Masters, K. L., Scarlata, C., Simmons, B. D., Smethurst, R. J., and Wright, D. (2019). Galaxy zoo: Probabilistic morphology through bayesian cnns and active learning. *ArXiv*, abs/1905.07424.

West, N. E. and O'Shea, T. J. (2017). Deep architectures for modulation recognition. *CoRR*, abs/1703.09197.

Wilkinson, D. M., Maraston, C., Goddard, D., Thomas, D., and Parikh, T. (2017). FIREFLY (Fitting IteRativEly For Likelihood analYsis): a full spectral fitting code. *Monthly Notices of the Royal Astronomical Society*, 472(4):4297–4326.

Appendix A

Code

A.1 Frameworks

```
from   astropy.io import fits
import glob
import numpy as np
import os
import pandas as pd
import pylab as plt
import seaborn as sns
import tensorflow as tf


from keras.models import Sequential, load_model
from keras.layers import Dense, Dropout, Flatten, Reshape, Conv1D,
MaxPooling1D, GlobalAveragePooling1D, Activation
from keras.optimizers import SGD, Adam, RMSprop, Adadelta
from keras.regularizers import l1
from keras.utils import to_categorical

from sklearn.metrics import classification_report, confusion_matrix

import keras.backend as K
```

A.2 Function definition

```
# root mean squared error (rmse) for regression (only for Keras tensors)
def rmse(y_true, y_pred):
    from keras import backend
    return backend.sqrt(backend.mean(backend.square(y_pred - y_true), axis=-1))

# mean squared error (mse) for regression  (only for Keras tensors)
def mse(y_true, y_pred):
    from keras import backend
    return backend.mean(backend.square(y_pred - y_true), axis=-1)

# coefficient of determination (R^2) for regression  (only for Keras tensors)
def r_square(y_true, y_pred):
    from keras import backend as K
    SS_res =  K.sum(K.square(y_true - y_pred))
    SS_tot = K.sum(K.square(y_true - K.mean(y_true)))
    return ( 1 - SS_res/(SS_tot + K.epsilon()) )

def min_max(x):
    return (x + np.min(x))/(np.max(x) - np.min(x))
```

A.3 Reading Data

```
Alhambra_data = pd.read_csv('../Alhambra/alhambra.csv', delimiter=',')
filters = ['id', 'objID',
           'Field', 'Pointing',
           'CCD',
           'RAdeg','DECdeg',
           'x','y','area','fwhm',
           'stell', 'ell',
           'a','b','theta',
           'rk','rf','s2n',
           'photoflag',
           'F365W','dF365W',
           'F396W','dF396W',
           'F427W','dF427W',
           'F458W','dF458W',
           'F489W','dF489W',
           'F520W','dF520W',
           'F551W','dF551W',
           'F582W','dF582W',
           'F613W','dF613W',
           'F644W','dF644W',
           'F675W','dF675W',
           'F706W','dF706W',
           'F737W','dF737W',
           'F768W','dF768W',
           'F799W','dF799W',
           'F830W','dF830W',
           'F861W','dF861W',
           'F892W','dF892W',
           'F923W','dF923W',
           'F954W','dF954W',
           'J','dJ',
           'H','dH',
           'KS','dKS',
           'F814W','dF814W',
           'F814W_3arcs','dF814W_3arcs',
           'F814W_3arcs_corr','nfobs','xray','PercW','Satur_Flag']

parameters = ['Stellar_Flag']

wavelengths = [3650,
               3960,
               4270,
               4580,
               4890,
               5200,
               5510,
               5820,
               6130,
               6440,
               6750,
               7060,
               7370,
               7680,
               7990,
               8140,
               8300,
               8610,
               8920,
               9230,
               9540,
               12500,
               16500,
               21900
              ]

np.random.seed(42)

Alhambra_data_shuffled = Alhambra_data.sample(frac=1).reset_index(drop=True)
```

A.4 Preprocessing

```
input_data = Alhambra_data_shuffled[filters]
input_data[input_data == -99] = 99
input_data = input_data.clip(0, 27)
output_data = Alhambra_data_shuffled[parameters]
output_data_norm = Alhambra_data_shuffled[parameters]


output_data_norm.loc[:,'zb_1']         = min_max(Alhambra_data_shuffled
                                                  .loc[:,'zb_1'])
output_data_norm.loc[:,'zb_min_1']     = min_max(Alhambra_data_shuffled
                                                  .loc[:,'zb_min_1'])
output_data_norm.loc[:,'zb_max_1']     = min_max(Alhambra_data_shuffled
                                                .loc[:,'zb_max_1'])
output_data_norm.loc[:,'tb_1']         = min_max(Alhambra_data_shuffled
                                                .loc[:,'tb_1'])
output_data_norm.loc[:,'Stell_Mass_1'] = min_max(Alhambra_data_shuffled
                                                .loc[:,'Stell_Mass_1'])
output_data_norm.loc[:,'M_ABS_1']      = min_max(Alhambra_data_shuffled
                                                .loc[:,'M_ABS_1'])
```

A.5 Training and Test sets

We show here the code for *zb_1*, but the code is the same for every parameter:

```
param = 'zb_1'

input_dim  = len(input_data.loc[0,:])
output_dim = len(output_data.loc[:, param])

train_perc = 0.10
test_perc  = 0.01

n_train = int(output_dim*train_perc)
n_test  = int(output_dim*test_perc)

NN_train_input  = np.array(input_data.loc[0:n_train, :])
NN_train_output = np.array(output_data_norm.loc[0:n_train, param])
                   .reshape(n_train+1, 1)
NN_train_output = np.power(NN_train_output, 0.1)

NN_test_input   = np.array(input_data.loc[0:n_test, :])
NN_test_output  = np.array(output_data_norm.loc[0:n_test, param])
                   .reshape(n_test+1, 1)
NN_test_output  = np.power(NN_test_output, 0.1)
```

A.6 Neural Network for *zb_1*

```
model = Sequential()

model.add(Dense(120, input_dim=len(filters), activation=’linear’))

model.add(Dense(1))
#model.add(Dropout(0.05))

sgd = SGD(lr=1e-12, decay=1e-6, momentum=1, nesterov=False)
model.compile(optimizer=’adam’, loss=’mse’, metrics=[r_square])
history = model.fit(NN_train_input, NN_train_output,
          epochs=150,
          batch_size=64,
          validation_data=(NN_test_input, NN_test_output),
          verbose=1)

model.save(’model_Zb1.h5’)
```

A.7 Neural Network for *stellarity*

```
model = Sequential()
n_Dense = np.linspace(24, 1, 2)

model.add(Dense(2, input_dim=len(filters), activation=’relu’))
model.add(Dense(3, activation=’relu’))
model.add(Dense(2, activation=’softmax’))

model.compile(optimizer=’adam’,
              loss=’categorical_crossentropy’,
              metrics=[’accuracy’])
history = model.fit(NN_train_input, NN_train_output,
          epochs=150,
          batch_size=16,
          validation_data=(NN_test_input, NN_test_output),
          verbose=1)

model.save('model_stellarity.h5')
```

Appendix B

First Project

During this project, the original considered goals were different to the ones in the late stages of the project. This happened as it turned out that they were too ambitious to be achieved within the temporal scope of the project.

This Appendix summarizes the goals, the fails and the successes identified during the early stages of the project, as a starting point for reaching the original goals with enough time and dedication.

B.1 Goals

The main objectives of the work were the following:

1. **To build a dataset for training and testing**, formed by **SSP!**s (**SSP!**s) with known parameters (kinematics, age and metallicity). The dataset will be formed by a large number of spectra of different signal-to-noise ratios (S/N or SNR, as this is a key factor for correctly deriving the stellar parameters) and arrays containing the parameters of the SSPs. The input arrays (spectra) will have a size of 1×4300, while the output arrays will have an estimated size of 1×1750[1]

2. **To design, train and test a Neural Network**, capable of accepting the spectra of the galaxies as the input, and returning a reliable array as close as possible to the parameters with which the spectrum was built. By means of a confusion matrix, we can estimate the accuracy of the Neural Network.

3. Once a high accuracy is obtained, the next goal should be to **define the range of spectral parameters** (in terms of spectral resolution, SNR, etc.) for which the Neural

[1] If the size of the network proves to be excessive for the computer available, there is the option of reducing both the sizes input and the output arrays.

network performs better.

4. **Establish a comparison** among the results obtained with different codes and the results obtained with the Neural Network.

5. **Apply the Neural Network to real well known galaxies**. This step will be of great usefulness to measure the performance of the Neural Network with real data, which may be significantly different from synthetic spectra.

B.2 Achievements

From the goals enumerated previously, the following could be achieved:

1. A large dataset was built from the MILES stellar population library (Falcón-Barroso et al., 2011). This dataset was built using data augmentation, creating custom transformations (simulating redshift, radial velocity, velocity dispersion, extinction by dust and emission by ionized gas). This transformations were applied to a low number of templates which were assigned a weight, normalized and added to simulate the spectrum of a galaxy.

2. A neural network was designed, trained and tested, being able to ingest and process the data.

B.3 Implementation

The first approach considered to achieve the goals was to try and estimate the parameters of the transformations applied to each one of the 350 templates in the MILES library. As there are 6 parameters for each one (redshift, radial velocity, velocity dispersion, extinction, gas emission and weight) the network has to derive 1,400 parameters.

Once this approach proved unreachable, the decision was to simplify the problem. A new set of data was created, but including only up to 4 different templates. With this approach, the number of parameters associated to each parameter is 8 (the same six as before, but we have to add the age of the stars contributing to the spectrum in the template, and the fraction of chemical elements relative to hydrogen). In this case, the network has to predict 32 parameters

B.4 Difficulties

While training and testing the neural network, some issues arised:

- As there are many input and output parameters, a large amount of crosstalk among neurons has been detected. This issue is amplified by the fact that the neural network was devoted to very different tasks at the same time. For instance, for a single template, the network has to predict: redshift, which ranges from 0 to 1; radial velocity, with values between -500 and 500 km/s; chemical abundances, which goes from 0 to 0.03...

- The input is too large. Considering that the spectra in MILES goes from 3540.5Å to 7409 Å with a spectral resolution of 0.9Å, the input array has a size of 4300. As the output size is 1400, the network necessarily consists of a very large number of parameters (in the order of 40 to 300 million). Therefore, it is necessary a huge amount of data to be able to build large enough training and testing datasets. It is estimated that a minimum of 500,000 synthetic galaxies are needed, which is a very time- and storage-consuming process.

B.5 New approaches

During the execution of the project, the two different approaches were incorrect, at least taking into account the temporal frame of the project. As future work, the following perspectives should be considered:

- Dividing the network in smaller networks. This point of view should be taken into account. The ideal situation would be to implement a number of networks equal to the number of outputs (1400). However, training, testing and using the model could be very computationally expensive, so a better approach would be to split the network in 6, one for each type of parameter to be predicted. A compromise solution would be to build 350 networks, one to predict the values associated to each of the templates.

- Using Invertible Neural Networks. These are a kind of networks which are trained using the input as output and vice versa. Once the network is trained, it is inverted (algebraically). This approach has the difficulty that the network must cover a series of requirements in order to be invertible.

- Using Generative networks. The idea behind this approach is to generate a synthetic spectrum with a set of parameters and compare the synthetic and the real spectrum. A reward function has to be defined, in order to maximize the function when the two spectra are identical.