Modeling the stellar spectrum of galaxies implementing dimensionality reduction and Machine Learning techniques

MÁSTER EN ASTROFÍSICA

Trabajo de Fin de Máster

Iñigo Irizar Loibide

Contents

List of figures ii

List of tables iii

1 Introduction 3
- 1.1 The study of galaxies . . . 3
- 1.2 Synthetic models . . . 3
- 1.3 Dimensionality reduction and Machine Learning techniques . . . 4
- 1.4 Objectives of this work . . . 6

2 Principal Component Analysis 8
- 2.1 Dimensionality reduction . . . 8
  - 2.1.1 Methods . . . 8
  - 2.1.2 Results . . . 9
- 2.2 pPXF . . . 12
  - 2.2.1 Methods . . . 12
  - 2.2.2 Results . . . 13

3 Neural Networks 19
- 3.1 Methods . . . 20
  - 3.1.1 Architectures . . . 20
  - 3.1.2 Activation functions . . . 22
  - 3.1.3 Normalization of the training set . . . 23
  - 3.1.4 Expanding to 3D and 4D . . . 23
- 3.2 Results . . . 24
  - 3.2.1 Training times: CPU versus GPU . . . 25
  - 3.2.2 Architectures . . . 26
  - 3.2.3 Activation functions . . . 27
  - 3.2.4 Normalization of the training set . . . 28
  - 3.2.5 Expanding to 3D and 4D . . . 28

4 Conclusions 36

List of Figures

1 Sample spectra from the SSP library created with PADOVA isochrones . . . 4
2 Principal components extracted from the SSP library 10
3 Reconstruction of the SSP library using PC 11
4 Relative contribution of each PC with regards to the spectral parameters . . 12
5 Kinematic maps of NGC 3608 obtained with pPXF 13
6 Fitting of the central galactic spectra using origonal library and PC extracted from the SSP library 14
7 Statistical parameters of the residuals of the central spectra fitting for 15 galaxies 15
8 Comparison of kinematic maps obtained for galaxy NGC 3608 using PC and SSP 16
9 Results of the differences for the kinematic properties after PCA 17
10 Scheme of a simple neural network 19
11 Representation of the propposed architectures 21
12 Scheme of a neural network suitable for both A_1 and A_2 21
13 Selection of available activation functions 22
14 Normalization of the training set after normalization of the dataset 23
15 Architecture used for the training time comparison between CPU and GPUs 25
16 Comparison between the accuracies of architectures A_1 and A_2 when trained with 78 SSP 26
17 Comparison of the results of architectures A_1 and A_2 27
18 Accuracy parameters resulting from training A_1 with 78 SSP through 1200 epochs 28
19 Comparison between the results obtained after 2 different data normalizations 29
20 Comparison of the accuracy of architecture A_1 after increasing the size of the training set 30
21 Comparison of the accuracy of architecture A_2 after increasing the size of the training set 31
22 Comparison of the accuracy of architecture A_2 after including all the IMF values 32
23 Accuracy of the learning process after including abundance ratio as parameter 33
24 Accuracy of the learning process for the full 4-dimensional spectral fitter . . 34

List of Tables

1 Naming conventions of the SSP library created with PADOVA isochrones . . 4
2 Time performance of pPXF code with SSP and with PC for 15 galaxies . . . 18
3 Times of the training for the spectral fitter: CPU versus GPUs 25

Abstract

Este Trabajo de Fin de Máster está enfocado a la implementación de métodos computacionales con el fin de reducir el coste temporal al correr códigos de ajuste espectral. Como objetivo secundario, nos proponemos implementar métodos de *Machine Learning* para facilitar el modelado de distribuciones espectrales de energía en espacios de parámetros N-dimensionales.

Estudiamos un método de reducción de dimensionalidad, en particular el método PCA, para tratar de reducir la dimensión de la base de modelos SSP que utiliza una rutina llamada pPXF para ajustar espectros galácticos. Convolucionando los modelos con parámetros cinemáticos, pPXF modela un espectro galáctico y lo compara con las observaciones, tratando de minimizar los residuos variando los parámetros. Trataremos de recuperar los parámetros cinemáticos de un grupo de galaxias utilizando un subconjunto de componentes principales extraídos del análisis PCA efectuado sobre dicha librería. Los resultados obtenidos con los componentes principales los comparamos con resultados presentados en la literatura.

La segunda parte del trabajo se centra en la implementación de Redes Neuronales para interpolar modelos SSP en espacios de parámetros N-dimensionales. Diseñamos distintas arquitecturas capaces de interpolar modelos SSP, por un lado construyendo un interpolador tradicional, y por otro lado creando una red que sea capaz de aprender las características espectrales de los SSP en función de sus parámetros físicos.

El análisis PCA se basa en el estudio estadístico de un grupo de datos, y trata de crear elementos ortogonles que expliquen la máxima variabilidad de los datos. Estos elementos forman un subespacio que conserva la capacidad explicatoria de los datos originales. El ejercicio de reducción de dimensionalidad lo aplicamos sobre una librería de SSP creada combinando la librería estelar MILES con isocronas PADOVA, y caracterizada por los parámetros edad y metalicidad. La librería posee un tercer parámetro, la pendiente de la IMF, pero es la misma para todos los elementos de la librería. Verificamos la utilidad del método propuesto reconstruyendo los elementos originales de la librería utilizando los componentes principales. Como último paso de este ejercicio, y con la idea de analizar la relación entre los componentes principales y los parámetros espectrales, seguimos un procedimiento descrito en la literatura. En las figuras presentadas en la sección de resultados del análisis PCA se puede ver con claridad que un subconjunto de 6 componentes principales es suficiente para reconstruir con precisión notable la librería original. En lo referente a la relación entre los componentes principales y los parámetros edad y metalicidad, encontramos que la edad está codificada mayormente en el primer componente principal, apareciendo sutilmente tanto en el segundo como en el cuarto. La metalicidad aparece más repartida entre el primer y segundo componente, y apenas contribuye en los demás elementos de la librería reducida.

Una vez completado el análisis PCA, pasamos a correr la rutina de ajuste espectral pPXF con los componentes principales generados en el ejercicio anterior. Lo primero a decidir es el número de componentes a utilizar. Para esto, ajustamos los espectros centrales de 15 galaxias utilizando diferentes números de componentes. Realizamos un análisis estadístico de los residuos para determinar que 10 componentes principales son suficientes para obtener ajustes satisfactorios de los espectros. Una vez hecho esto, procedemos a calcular los mapas cinemáticos para las 15 galaxias con los componentes principales, obteniendo resultados similares a los presentados en la literatura. El ahorro

temporal resultante de utilizar una librería reducida es considerable, como comentamos en la sección de conclusiones.

Para la segunda mitad del trabajo, hemos hecho uso del paquete *Keras* como herramienta de construcción y entrenamiento de Redes Neuronales.

Para comenzar, proponemos dos tipos de arquitecturas, pensadas para interpolar modelos SSP en un espacio de parámetros bidimensional, definido por la edad y la metalicidad. Estudiamos el efecto que puede tener la elección del tipo de normalización de datos, interpolando una mitad de la librería de SSP utilizada anteriormente después de haber entrenado la red con la otra mitad. Ambas arquitecturas resultan ser capaces de interpolar en el plano definido por la edad y metalicidad de los SSP, como se observa en los resultados. Una vez demostrada la validez de las redes a la hora de interpolar SSP, pasamos a aumentar la complejidad del problema añadiendo más parámetros. Con este ejercicio, aumentamos la dimensionalidad del espacio de parámetros en el que trabajarán las redes. En este caso, hacemos uso de tres librerías, que a diferencia de la librería utilizada en el análisis PCA, están caracterizadas por cuatro parámetros espectrales. Las tres librerías se componen de SSP repartidos en 12 metalicidades, 53 edades y 14 pendientes de IMF, y la diferencia entre ellas radica en el cuarto parámetro, el ratio de abundancia.

Haciendo uso de estas tres librerías construímos una serie de 5 redes, que van aumentando en complejidad. Las primeras 2 redes las diseñamos para ver el efecto que tiene sobre los resultados el entrenar con una librería, para ambas architecturas. La tercera red la utilizamos para estudiar el efecto de incluir un tercer parámetro espectra. Las redes 4 y 5 sirven para introducir la cuarta dimensión en forma de ratio de abundancia. Del el ejercicio propuesto para las 5 redes, obtenemos resultados de interpolación con una precisión notable en la mayoría del espacio de parámetros en cada caso. Sorprendentemente, encontramos que, a medida que el problema se hace más complejo, la precisión de las interpolaciones no disminuye. Esto nos lleva a pensar que las redes neuronales aprenden con mayor eficiencia si se entrenan con bases de datos más extensos.

La implementación de redes neuronales para interpolar modelos SSP en parámetros N-dimensionales resulta una mejora respecto a métodos tradicionales en cuanto a que, una vez entrenada la red, la interpolación es instantánea independientemente de la dimensionalidad del espacio de parámetros.

A lo largo de este trabajo desarrollamos dos métodos que pueden llevar a un ahorro en el tiempo de computación de códigos de ajuste espectral como pPXF.

1 Introduction

1.1 The study of galaxies

The study of galaxies is a relatively young field of research and has been subject to a huge thrust due to the technological advance. Learning about the external galaxies help us understand better the formation and evolution of the Milky Way. The study of external galaxies also provides with the opportunity to examine the time evolution of the universe, and to learn how the physical theories adjust to the observational data.

The information regarding external galaxies comes in the form of light. This light is detected by the telescopes, and contains *photometric* and *spectroscopic* information. Apart from describing the morphology of the objects by the distribution of light on the sky plane, the photometric data is often used to determine stellar and galactic magnitudes, which characterize parameters such as temperatures and distances. Combining both of these characteristics, more information regarding the interestellar and intergalactic dust may be obtained, for example.

The spectroscopic information is encoded in the spectra of the incoming light, which, by definition, is the spectral energy distribution of the light. This distribution contains information of the atomic and molecular populations from where the light was originated. Measuring the relative depth and width of absorption lines, it is possible to extract information about element abundances, superficial gravity and rotational velocity, in the case of the stars, and populations, in the case of galaxies. In addition to that, the measured spectral lines are sensitive to the local dynamics, as absorption and emission lines *shift* towards red or blue, depending on the relative motion between the emitter and the observer. Thus, analyzing galactic spectra, we can obtain information regarding the dynamics and populations of galaxies.

1.2 Synthetic models

The galaxies appear projected on the sky plane, but the light originates in a 3-dimensional distribution of stars. In fact, the signal recieved on each resolution element of the detectors is the sum of the light of the stars across the *Line Of Sight* (LOS).

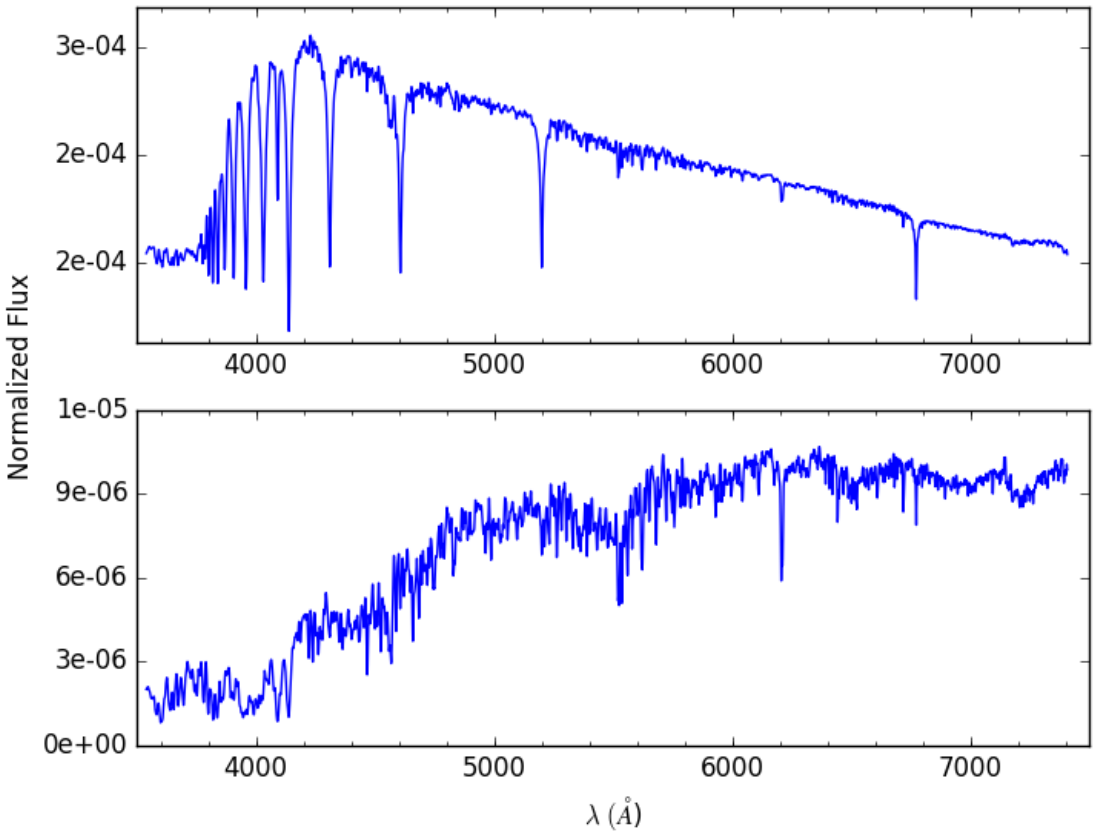
Most of the methods available to extract information from galactic spectra consist in combining synthetic models, in order to construct the LOS that most accurately fits the observed data. This is done convolving single-aged stellar population (SSP) models with kinematic parameters that describe the motion of the stars along the line of sight. The SSPs, available on the web[1], are created combining the MILES stellar library, containing around 1000 stellar spectra, with solar-scaled PADOVA isochrones, as explained by Vazdekis et al. [2010]. The resulting library gathers 156 SSP, with ages between 0.063 and 17.7828 Gy, and a spectral range extending from 3525 to 7500 Å. In this case, the parameter space is two-dimensional, since all the spectra have the same unimodal IMF slope of value 1.30. The 156 SSP are distributed on 26 age and 6 metalicity values.

The SSP filenames encode the information of the spectra, regarding different parameters. We explain the naming convention on the following table.

[1] ftp://miles.iac.es

Filename	IMF	Age (Gy)	$[M/H]$
Mun1.30Zm1.71T01.0000.fits	1.30	1.0000	-1.71
Mun1.30Zp0.22T17.7828.fits	1.30	17.7828	0.22

Table 1: Naming conventions of the SSP library created with PADOVA isochrones. The filename contains the information about the age, metallicity and IMF slope value.



(a)

Figure 1: Representation of the spectroscopic data found in the files presented in table 1. The top panel spectra corresponds to an age of 1 Gy and a solar-scaled metallicity of -1.71. The age and metallicity of the bottom panel are 17.7828 Gy and 0.22 respectively.

The differences between the two samples presented in figure 1 are clearly visible at first sight. The young and metal poor has a much lower spectral line density, and the Balmer series are of the most important features of the spectra. On the contrary, the old and metal-rich sample shows a high spectral line density, as expected from the stellar evolutionary theory, while the Balmer series is barely noticeable.

1.3 Dimensionality reduction and Machine Learning techniques

It is often the case where the datafiles containing extragalactic observations turn out to have large size, and the implementation of complex mathematical procedures involved in

spectral fitting techniques results computationally expensive. Therefore, the implementation of dimensionality reduction methods becomes a highly advisable practice.

A variety of methods have already been implemented with such purposes, proving their utility when compressing the information contained in large datasets into a much smaller subsets. Some of the most popular methods are Independent Component Analysis (ICA), a slightly modified version of ICA named Ensamble-Learning ICA (EL-ICA), and Principal Component Analysis (PCA).

The ICA method performs a statistical analysis of a multidimensional dataset and creates independent components capable of describing such dataset. Lu et al. [2006] performed an exercise involving EL-ICA, where they apply the method to a set of 1326 SSP, and utilize the resulting independent components as templates to model synthetic galactic spectra. An *Ensemble-Learning* algorithm is applied to the ICA, in order to avoid overfitting solutions. After performing the EL-ICA, the authors found that the SSP library could be consistently reduced to a set of 6 IC. As discussed in the article, the physical meaning of the obtained independent components is closely related to the physical parameters defining the spectral library, thus they classify the IC according to their spectral type, since they turn out to be similar to stellar spectra. Afterwards, the authors use the calculated independent components to model galaxies from SDSS, being able to derive their spectral parameters.

Making use of another of the dimensionality reduction methods, Chen et al. [2012] perform the PCA over a set of 25000 star formation histories (SFH) created using SSP. After obtaining the principal omponents (PC), they determine that 7 principal components describe accurately the SFH library, and project the data on the space that define the PC. In order to determine the physical parameters, the authors propose a method based on the similarity between the library elements, the SFH, and the galaxies on the reduced PC space, which is calculated comparing their weights on the PC space. As a result, the researchers show that PCA is an efficent way to significantly reduce the model libraries used for modelling galactic spectra, as well as the fact that it is possible to recover the information about the physical parameters encoded in the original library from the reduced PC set.

In a similar exercise, Ronen et al. [1999] applied PCA analysis to 1850 age-dependent galactic spectra, formed by assuming star formation history SFH, initial mass function IMF and metallicity. The study determines the correlation between the PC and the age of the modelled galactic spectra. In addition to that, after forming a second set of spectra by sampling more metallicity values, they study the effect of metallicity on the principal components, which results in a higher variability on the 2^{nd} PC. By plotting the PC weights alongside the spectral parameters, the authors reveal that the parameters of modelled galaxies may be explained using a small set of principal components.

In addition to the techniques mentioned above, clustering algorithms prove to be effective when determining characteristic features of a given dataset. *KMeans* is one of such algorithms, which divides the data in k groups, centered in k corresponding centroids on the original parameter space, instead of proyecting the data onto a different space, as PCA and ICA do. The centroids of the groups correspond to labels, which can be used as *classes* inside the dataset. Using this method, Almeida et al. [2010] were able to automatically classify galactic spectra from the Sloan Digital Sky Survey (SDSS).

KMeans can be considered as an unsupervised learning algorithm for which, in contrast to more sophisticted techniques, the learning process aims to the convergence of the calculated centroids. Since the begining of the 90's, a branch of computing known as *Machine Learning* has gained popularity among data scientists. In particular, the so called *Neural Networks*

(NN) are often used with purposes such as classification of data, time-dependent model predictions and non-linear problem solutions.

Neural Networks have been implemented to solve problems on diverse fields, starting from image classification and pattern recognition, as proved by Rawat and Wang [2017], to bilinear curve fitting for robotics applications, as shown by Mital and Chin [1998], to create positioning controllers. Our work is focused on finding solutions of curve fitting problems, similar to the work of Bishop and Roach [1992], so the complexity of task is much lower than the ones mentioned here.

The most popular application of NN consist in pattern recognition in images with classification purposes. In this case, *Convolutional Neural Networks* are trained to codify an image through a series of convolutional and fully connected layers. The result of this codification is then associated with a previously defined *class*, teaching the network to associate similar outputs with the mentioned class. An introductory example of an implementation of NN for digit recognition may be found on the internet[2], where MNIST database is used to teach a NN to classify images of handwritten numbers.

Using a similar principle, Ramos et al. [2017] made use of Deep Learning techniques to determine horizontal velicity fields in the solar chromosphere. Making use of a mix of convolutional and dense layers, they are able to determine the horizontal velocity field of the cromosphere of the Sun, which would otherwise be impossible, since horizontal velocity is not subject to spectroscopic doppler effects.

Mimmicking the image classification problem, Hála [2014] showed that convolutional neural networks are capable of classifying 1-dimensional data corresponding to spectral energy distributions of astrophysical objects. Following the same procedure as when classifying $2D$ images, they designed a network capable of discerning among spectra of stars, galaxies and quasars.

Neural networks also have proved to be able to solve regression type problems, like parametrical curve fitting. Making use of a configuration of fully connected layers, also known as *Multi-layered Perceptron* (MLP), Bishop and Roach [1992] designed a network capable of determining in real time the parameters of the spectral lines generated on a nuclear fusion reactor.

1.4 Objectives of this work

With the aim of decreasing the computational cost when studying dynamics and populations of galaxies using spectral fitting methods, the objective this work consists on the implementation of both dimensionality reduction and Machine Learning techniques. The first half will focus on the application of the PCA on the SSP library mentioned earlier, and using the resulting PC to recover kinematic parameters of galaxies presented in the work of Emsellem et al. [2004] using a spectral fitting code. The aim of this exercise is to prove that a dimensionality reduction technique maintains the information encoded on the original library unaltered, but speeds up computation.

The second half of the work, grounded on the context of Machine Learning, will consist on building an interpolator of SSP. This interpolator will be designed to return a SSP model corresponding to a pair of parameters (age,metallicity) on the 2D space described on subsection 1.2. The task will be approached from two different points of view. First, we will design

[2] https://machinelearningmastery.com/handwritten-digit-recognition-using-convolutional-neural-netwo

and train a traditional interpolator, which will return the weights of the desired new element on the base formed by the SSP. The second approach will consist on training a network to return the entire SSP matching the target (age,metallicity) pair, learning the pixel-to-pixel or wavelength variations as a function of the input parameters. Afterwards, we will attempt to extend the method to higher dimensions, including parameters like IMF slope and abundance ratio [α/Fe].

The advantage of the NN compared to the traditional interpolators resides that, once trained, the prediction of any SSP belonging to the parameter space in which the NN was trained is instantaneous independently of the dimensionality of the parameter space, in contrast to the time need by conventional interpolation algorithms.

The following sections will focus on the theoretical background and methods applied to achieve the goals mentioned above, as well as the results obtained from the implementation of the described time-cost reduction techniques.

2 Principal Component Analysis

This chapter is focused on the methods and tools used to perform the dimensionality reduction by implementing PCA, as well as the results obtained from the analysis. The procedure was implemented upon the library described in section 1.2, which contains the SSP models. After PCA was performed, the reduced library was tested by running a spectral fitting code named pPXF and analizing the accuracy of the set of PC when comparing the results obtained with the ones presented in the work of Emsellem et al. [2004]. For that, we picked a group of galaxies from *SAURON* survey of early-type galaxies, as subjects for the pPXF routine.

2.1 Dimensionality reduction

Principal Component Analysis (PCA) is a method that provides means of reducing large datasets into more manipulable objects, while maintaining the explanatory capability of the original set. Once performed, the result of the analysis comes in form of vectors, the Principal Components (PC), that have the same structure of the dataset items, but different meaning.

A simple and descriptive approach of the Principal Component Analysis may be found in the web[3] in the form of a tutorial. This tutorial proves to be an excelent introduction to the problem, explaining with detail the basic concepts of the implementation of PCA on a set of data, and provides the reader with a guided step-by-step practical example. As pointed further in this section, there are fully-functional codes available that perform the analysis, however the tutorial found online is a worthwhile lecture to rapidly understand the purpose and mechanics of PCA.

2.1.1 Methods

The PCA projects the original n-dimensional dataset into the *feature space*, which is a d-dimensional subspace with $d < n$, originated by the PC. The dimensions, or axis, of the feature space represent the highest variability of the dataset, being the main axis, or the first PC, the vector that maximizes the variance of the dataset. Apart from dimensionality reduction, PCA is a tool often used to extract characteristic features of the dataset. The case of interest on this work consists in a dimensionality reduction of a spectral library, so the principal components are expected to be similar to spectral energy distributions, despite not beeing strictly so.

The analysis is performed by calculating the eigenvalues and eigenvectors of the scatter matrix generated from the dataset. Let us name x_k the dataset. The scatter matrix S can be calculated as follows:

$$m = \frac{1}{n}\sum_{k=1}^{n} x_k \tag{1}$$

$$S = \sum_{k=1}^{n}(x_k - m)(x_k - m)^T \tag{2}$$

[3] http://sebastianraschka.com/Articles/2014_pca_step_by_step.html

The eigenvectors and eigenvalues are calculated by solving the equation

$$S\nu = \lambda\nu \tag{3}$$

Here ν refers to the eigenvectors, the principal components, and λ are the eigenvalues. This eigenvalues represent the amount of variance explained by each principal component ν. The PC with a highest λ contributes the most describing the variance of the dataset.

There are several packages that provide tools to perform the PCA. As the algebra needed for the method is not too complicated, it is relatively straightfowrard to create an original code to perform the analysis. However, for the sake of efficency, we decided to use the already existing *SKlearn* package from SciKit[4]. This tool returns the principal components, the original data reconstructed using PC, and the explained variance ratio of each PC, among many other features. This ratio is returned in form of a vector, and it turns out to be a useful parameter in order to determine the reduced dimension d. Principal components that present a low explained variance ratio may be discarded when forming the reduced subspace.

As the physical meaning of the principal components may not be clear at first sight, [Chen et al., 2012] propose a method to study the relation each PC has with the parameters that describe the library. The procedure consists in defining a quantity conformed by the projections of the original spectra in the PC space and the spectral parameters of the library, and minimizing that quantity.

$$\Delta = \sum_{i=1}^{n} \left[\sum_{\alpha} X_\alpha C_{i,\alpha} + Z - P_i \right]^2 \tag{4}$$

On equation (4) X_α and Z are the parameters that minimize Δ, which reflect the combination of PC and the *zeropoint* that effectively describe the parameter P of the library. The matrix $C_{i,\alpha}$ is the amplitude of the α^{th} PC for the i^{th} template, and is calculated in the following way:

$$C_{i,\alpha} = \sum_{\lambda} (X_{i,\lambda} - m_\lambda) E_{\alpha,\lambda} \tag{5}$$

Once Δ is minimized, the contribution of the α^{th} PC estimating the parameter P is:

$$P_{pc}(\alpha) = \frac{\sum_{i=1}^{n} |X_\alpha C_{i,\alpha}|}{\sum_\alpha \sum_{i=1}^{n} |X_\alpha C_{i,\alpha}|} \tag{6}$$

After the PCA analysis is performed, we will feed the pPXF code with the resulting principal components, with the aim of reproducing some of the results obtained using the original library.

2.1.2 Results

First, the first 3 principal components, resulting from the PCA performed on the PADOVA library, are shown in Figure 2. As discussed before, the PC are formally equal to spectral energy distributions; however, some of the components present negative flux. This is not, by any means, possible if the object is expected to be a real spectra.

[4] http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html

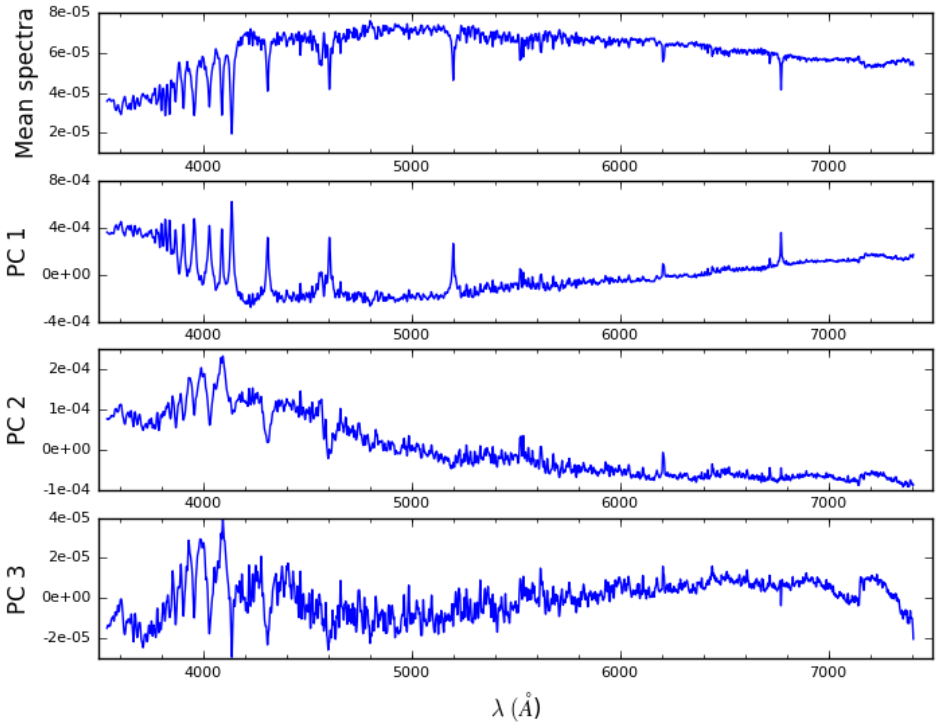


Figure 2: First 3 principal components extracted from the SSP library, alongside the mean spectra.

In principle, the PC represented in figure 2 hold relation with the spectral parameters age and metallicity, despite not being clear at first sight. For example, it is reasonable to say that the 1^{st} PC is related to the age, since the Balmer series is a prominent feature of the eigenspectra, and Balmer lines may be considered as age markers in stellar and galactic spectra. The relations in regards to metallicity seem even more difficult to determine, unless a detailed scrutiny of spectral lines is performed. However, as the section progresses, an alternative method will be tested in order to further determine the relations between the spectral parameters and the principal components.

The next step consisted in reconstructing the original library using the PC found on Figure 2. On Figure 3 we show an example of a reconstructed SSP alongside its original counterpart, followed by the reconstruction accuracy on the whole age/metalicity plane.

Analyzing both images shown in Figure 3 it is safe to say that the dimensionality reduction was succesful. The residual of the plot on top is negligible compared to the spectra itself. Moreover, on the bottom plot it can be seen that the overall standard deviation of the residual does not exceed the 2%, having its maximun on one of the boundaries of the parameter space. As an additional feature, it is possible to infer the effect of the sampling of the library on the bottom panel of Figure 3. The metalicity axis presents a smoother variation, in comparison with the axis representing age. This is so because the age grid has a higher resolution than the metalicity grid.

On an attempt to deduce how the principal components are related to the parameters of the library, we followed the steps shown by Chen et al. [2012], using equation (6).

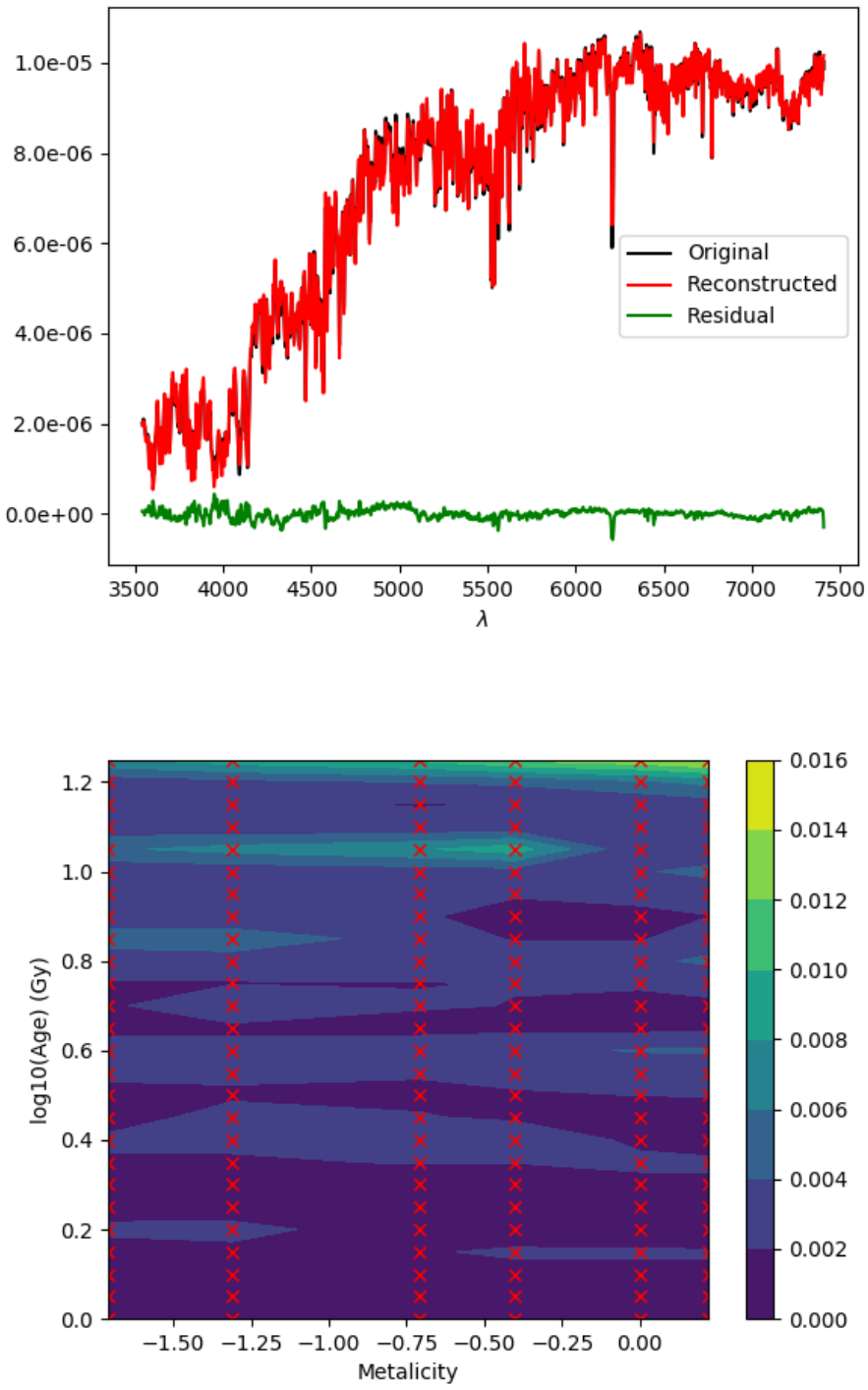


Figure 3: Reconstruction of the SSP library using 6 PC. On the top panel we show reconstruction of a sample SSP on arbitrary flux units, corresponding to the *top-right* cross on the bottom panel. On the bottom panel we plot the reconstruction accuracy on the 2D parameter space. The crosses show the points corresponoding to the SSP on the age-metallicity plane. The values encoded in the colorbar correspond to the standard deviation of the normalized residual of the reconstruction.

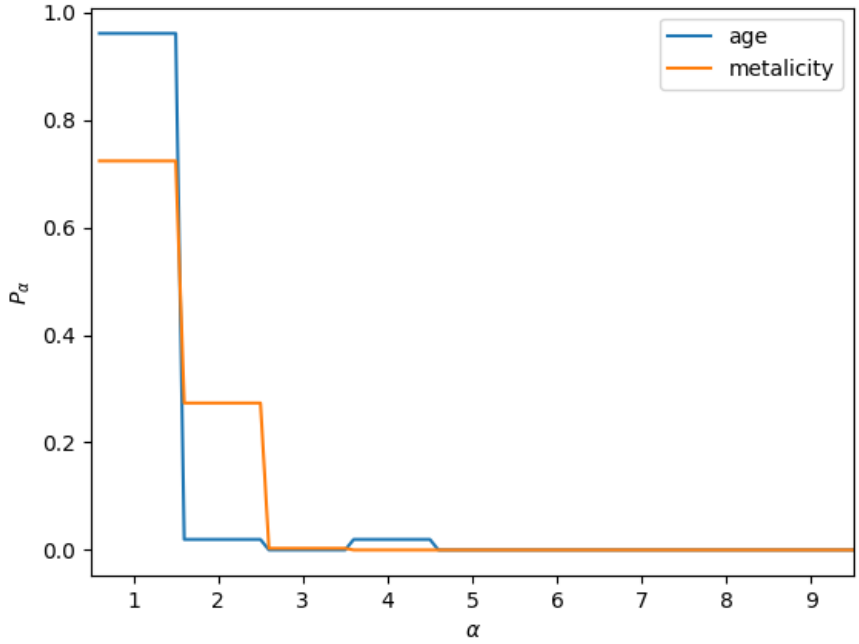


Figure 4: Relative contribution of the first 9 PC to the explanation of age and metalicity. The values were calculated using equation (6).

In figure 4, the plot represents the relation of each PC in regards to the spectral parameters age and metallicity. The results obtained from equation (4) show that age is mainly encoded on the 1^{st} PC, while being slightly present in PCs 2 and 4. The metallicity appears to be less related to PC1 than age, while its significantly present on the 2^{nd} component. This comes to agreement with what we deduced from looking at Figure 2.

2.2 pPXF

After succesfully completing the principal component analysis for the SSP library, the following step consisted in the analysis of the results obtained from the pPXF routine when feeding it with the PC, instead of the SSP. Our study covered kinematics analysis, and attempted to reproduce the results obtained with SSP models, which may be found in the work of Emsellem et al. [2004].

2.2.1 Methods

pPXF is a spectral fitting code, created by Cappellari and Emsellem [2004] which provides means to extract information of the kinematic and population properties of a galaxy, starting from a set of single stellar population models.

The spectral data of the galaxy of study comes in the form of a datacube, a typical output of a Integral Field Unit. This device divides the field of view in sections, and generates a dataset containing the spatial data in $2D$, while keeping the 3^{rd} dimension for the spectral information. The spatial part contains the distribution of light projected on the sky plane,

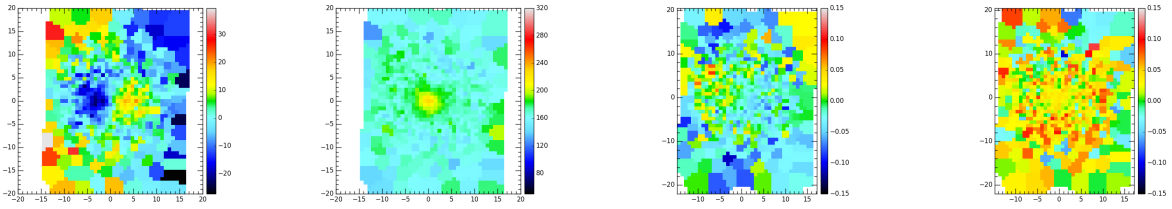
and the spectral part contains the spectral energy distribution of each resolution element on the 2D plane.

The spectra stored over the 3^{rd} dimension of the datacube is generated in the galaxy by adding the light of all the stars along the Line Of Sight (LOS) that crosses the galaxy in a 3D volume. Therefore, it is the addition of light coming from a large number of stars. pPXF convolves a set of templates with a theorethical Line Of Sight Velocity Distribution (LOSVD), builds a model spectra and compares it to the observations. The lines of sight are generated using gauss-hermite polynomials, characterized by the *mean velocity* μ and the *velocity dispersion* σ. The possible asymetries are added by integrating two parameters, h_3 (*skewness*) and h_4 (*kurtosis*). Thus, the line of sight is characterized by 4 parameters: μ, σ, h_3 and h_4.

As the galaxy of study is a relatively faint object, a threshold for the signal-to-noise ratio is set to obtain reliable and consistent results. In order to achieve this, the sky-plane is redistributed by the method called *Voronoi Binning*[5]. The binning sums the photometric fluxes around a coordinate on the sky-plane that corresponds to a measured spectra, covering the area needed to obtained the threshold S/N.

The kinematic parameters, namely μ, σ, h_3 and h_4, characterize the LOSVD corresponding to each of the bins created by the Voronoi Binning, and are directly estimated by pPXF. Thus, kinematic maps may be generated portraying the parameters that provide the best fit for the galactic spectra of each of the bins.

As an example, we present the results of the execution of the pPXF routine in Figure 5, with data from of the galaxy NGC 3608, and using the SSP library presented in section 1.



(a) (b) (c) (d)

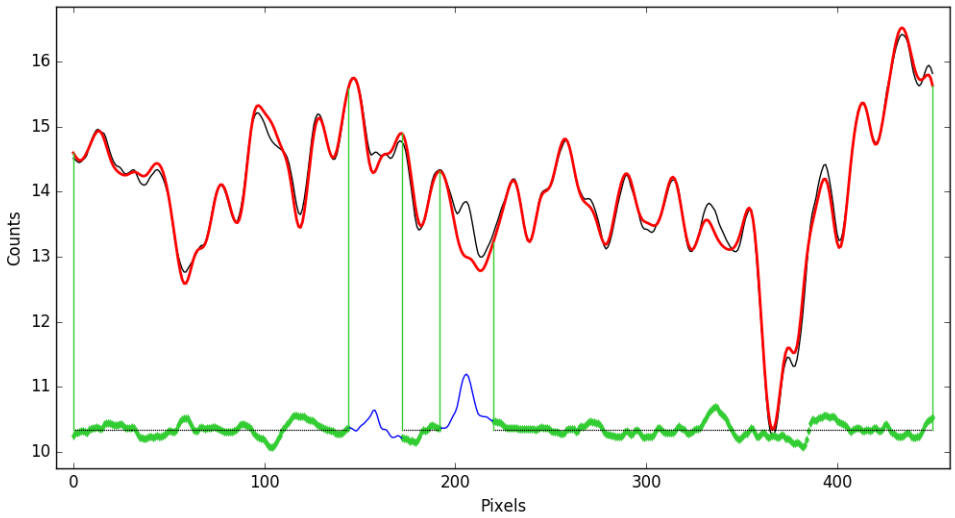
Figure 5: Kinematic maps of NGC 3608 obtained with pPXF, using the SSP library. The subplots represent (a) the mean velocity μ in (km/s), (b) the velocity dispersion σ in (km/s), (c) h_3 and (d) h_4.

2.2.2 Results

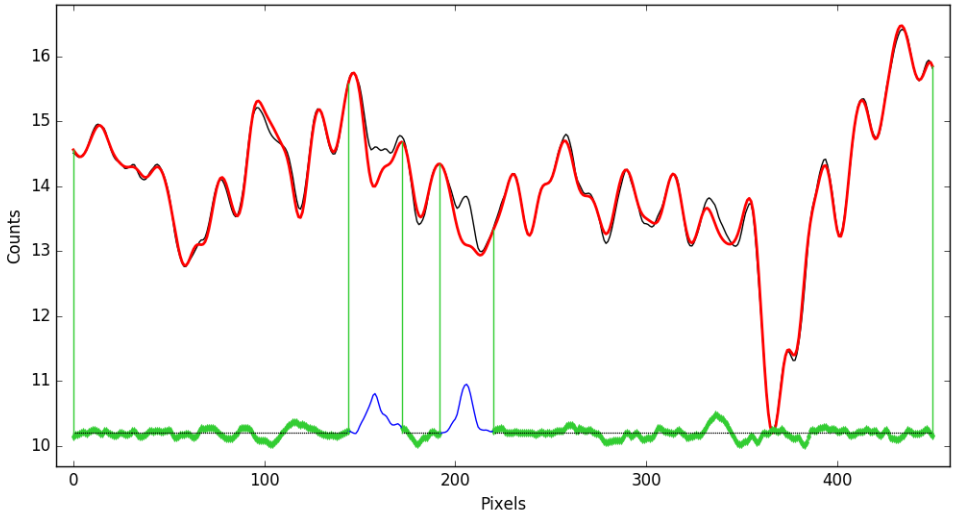
Making use of the principal components obtained from the dimensionality reduction exercise, we carried a kinematic analysis for galaxy NGC 3608, in order to test the valitidy of the PC when fitting real galactic spectra.

The next figure shows a comparison of the fitting for the spectra corresponding to the galactic center, where the signal-to-noise ratio is highest.

[5] http://www-astro.physics.ox.ac.uk/~mxc/software/



(a)



(b)

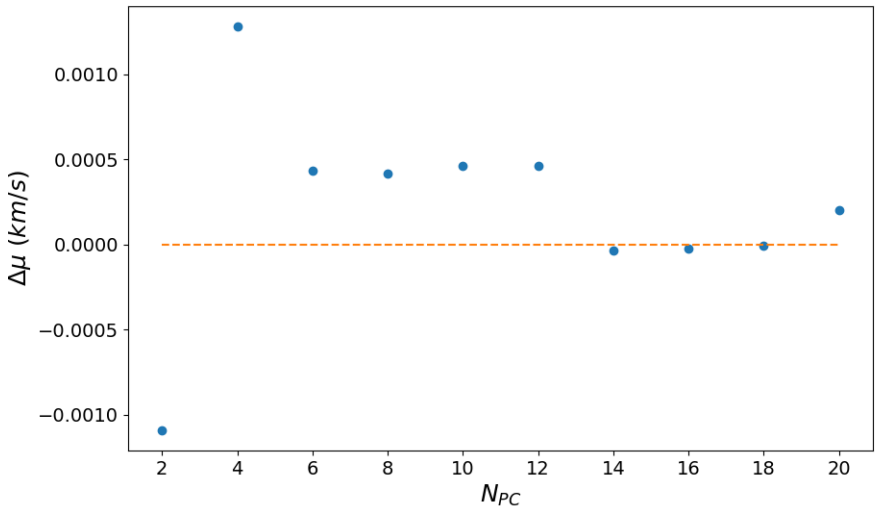
Figure 6: Central spectra of NGC 3608 fit using (a) the SSP library and (b) 10 principal components extracted from the models.

Figure 6 shows that the spectral fitting is slightly improved feeding the code with 10 of the principal components extracted from the SSP library, instead of the 156 elements comforming the library. As mentioned in section 2, the PC represent the axis holding the highest variability regarding the data. Taking into account the fact that the library was created combining stellar spectra and isochrones, it is plausible that some elements of the library repeat spectral features. The fact that the spectral fitting gets improved using principal components, which maximize the explanation variability of the data, is somehow a result that could be anticipated.

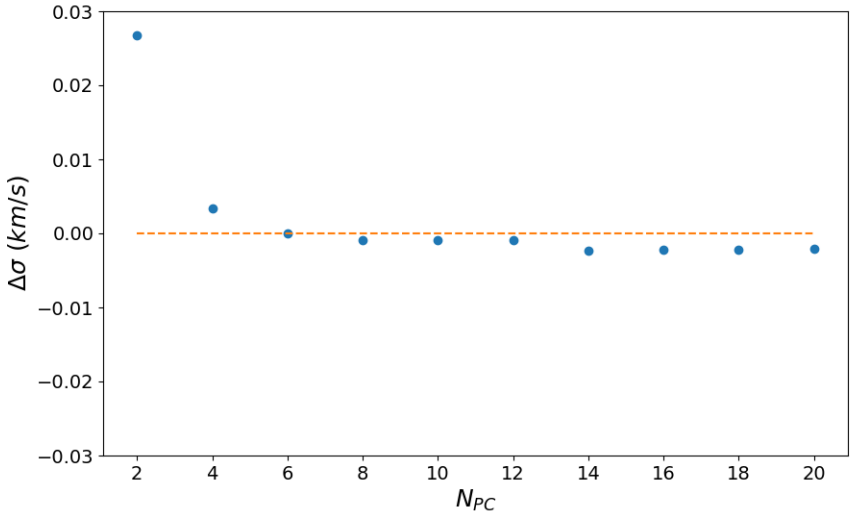
However, a crucial factor for the dimensionality reduction to be practical is the number of PC that will form the reduced subspace. Should the number of elements needed be of

the same order of magnitude, the implementation of the PCA with means to optimize time costs would be pointless. The minimum number of PC capable of reproducing the kinematic properties has been determined on the following way.

We performed a fit for the central spectra of 15 different galaxies feeding pPXF with 2 to 20 PC, in steps of 2 and compared the results to the fits obtained using SSP. The statistical parameters μ and σ of the residuals, which are representative the accuracy of the fitting, are shown in figure 7.



(a)



(b)

Figure 7: Statistical parameters of the residuals of the central spectra fitting for 15 galaxies with different numbers of PC. The top panel shows the mean value μ, and bottom panel shows standard deviation σ.

As can be seen in Figure 7, the spectral fitting improvement stops between 6 and 8 PC,

therefore it makes no sense to utse 20 principal components. This is noticeable as μ starts fluctuating after 8 PC. However, by visual inspection of the kinematic maps resulting from using such a low number of PC indicated that, even if the central spectra fit stops improving after 8 PC, more principal components where needed to successfully recover the kinematic parameters.

On the bottom panel we see that the dispersion of the residuals continues decreasing, but this might be a signal of reaching overfitting of the central spectra. Therefore, we use the mean of the distribution of the residuals, plotted on the top panel of Figure 7 in order to choose the number of PC to use.

Once the number of PC to use was determined, a complete kinematic analysis for galaxy NGC 3608 was performed using 10 principal components. In figure 8 we show a comparison depicting the results obtained using both PC and SSP.

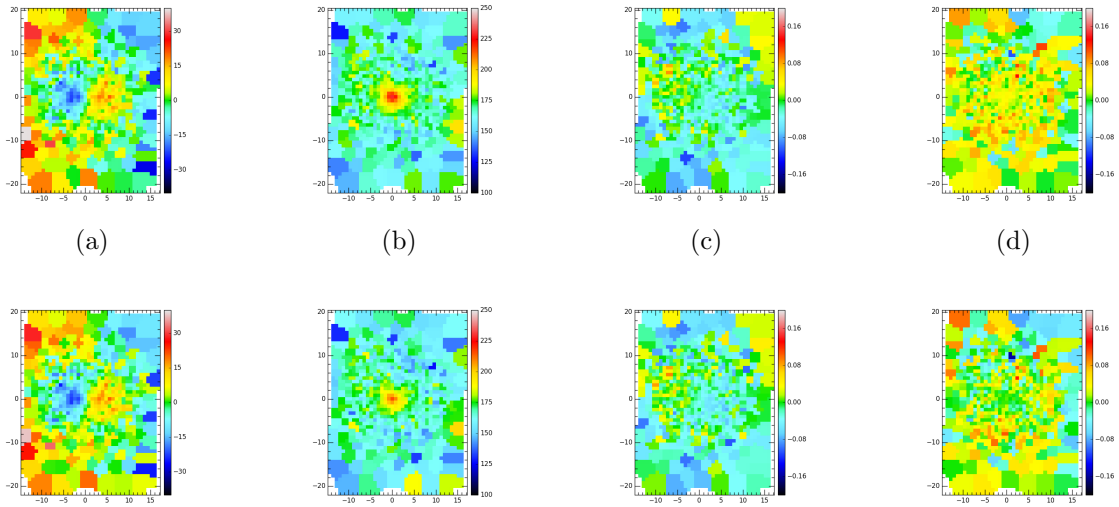
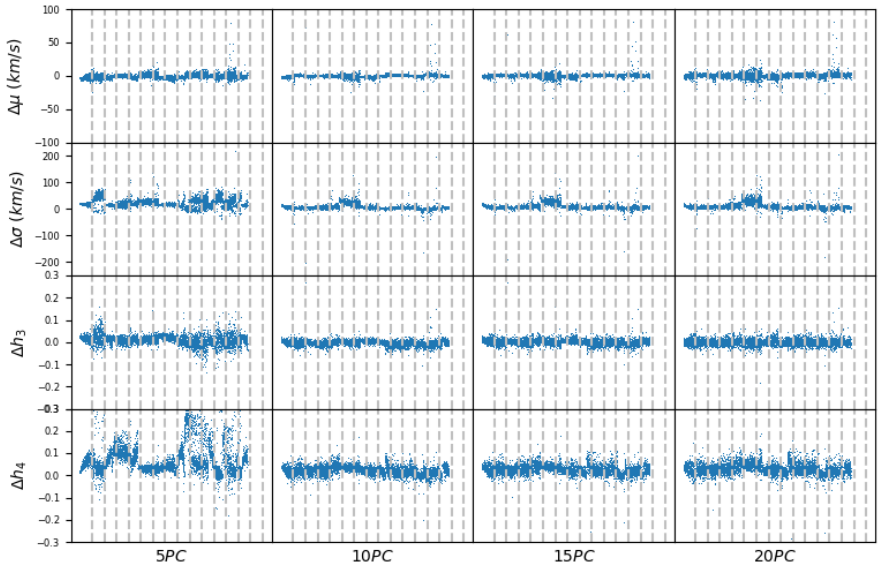
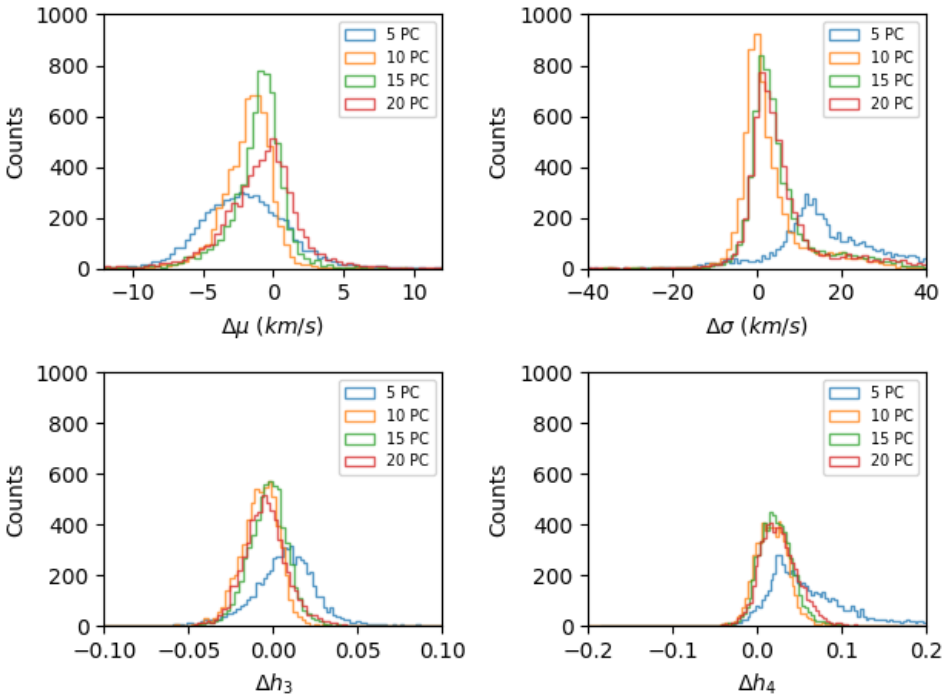


Figure 8: Kinematic maps obtained for galaxy NGC 3608 by running pPXF. The images on the top row result from using the SSP, and the bottom row corresponds to the maps obtained with a subset of 10 PC. The magnitudes correspond to (a,e) mean velocity μ in km/s, (b,f) velocity dispersion σ in km/s, (c,g) h_3 and (d,h) h_4.

Simultaneously, with the aim of supporting the choice of PC number, the residuals of the comparison between the kinematic maps found in the literature and the ones obtained with 5, 10, 15 and 20 PC were calculated. In thiscase, these residuals are a direct representation of the similarity between the maps obtained with PC and with SSP.



(a)



(b)

Figure 9: On the top panel, we show dispersion plots of the differences between the kinematic maps obtained after feeding pPXF with 5, 10, 15 and 20 PC. The 15 galaxies for which the exercise was performed are divided by vertical lines. On the bottom panel we present the same magnitudes in form of histograms.

The results shown in Figure 9 reassure the choice made for the number of PC to be used,

as no significant improvement is achieved with more PC. The major improvement shows transitioning from 5 to 10 PC. Even if single galaxies may show slightly smaller residuals with a higher number of components, 10 components show to be capable of obtaining results comparable to the ones obtained using SSP.

On table 2 it is clearly visible that the time cost is drastically reduced when running pPXF with principal components. Thus, we confirm that the Principal Component Analysis proves to be an effective method when reducing the time cost of spectral fitting routine pPXF.

Library	Total time
PC	31 m 45 s
SSP	542 m 37 s

Table 2: Time performance of pPXF code with SSP and with PC for 15 galaxies.

As an ending of the chapter, we conclude that principal component analysis turns out to be a satisfying tool in regards with time optimization of spectral fitting codes like pPXF, since the information concerning kinematics can be recovered with a much smaller time cost. PCA proves to be able to reduce the dimensionality of a SSP library from 156 elements to a subset of 10 principal components maintaining the explanatory power of the original dataset.

3 Neural Networks

As discussed in the introduction, Neural Networks are often used to build predictive models and perform tasks like image classification and pattern recognition. The ability to perform such jobs depends on two main features: the *architecture* of the network and the *training set*. The former feature defines the task for which the NN is built. For example, an image classifier needs to have multiple layers for operations like convolution and pooling, whereas a logical NN that represents XOR operation only needs two layers, one receiving the *input* and one returning the result.

The base component of a neural network is the *layer*. A layer is a unit that performs an operation upon the *input* which has been feed with. A simple neural network is composed of an *input layer*, an *output layer* and one or more optional *hidden layers*.

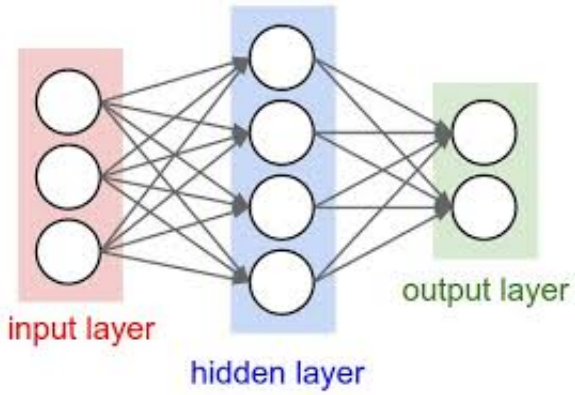


Figure 10: Scheme of a basic neural network containing an input layer, a hidden layer and an output layer. The network recieves a three-dimensional input and returns a vector of 2 components. Each neuron of the hidden layer is conected to all the neurons from the previous and next layers, forming a configuration known as *fully connected* network.

In figure 10 the circles represent the *neurons*. Each neuron processes the inputs coming from the previous layer, before transmittimg the output to the next one, on the following way:

$$Y = R\left(\sum_i X_i w_i + b\right) \tag{7}$$

Performing the operations described above, a neuron returns the output Y from the input X, multiplying with weights w_i and adding a bias b. The R refers to the activation function, which has to be chosen according to the task of the NN. The most used activation functions are the *softmax* and $ReLU$, which are implemented when teaching a network to solve non-linear problems. However, the use of these functions requires a preprocessing of the training set, since both activations restrict the value range of the outputs of the neurons to a range between 0 and 1. An example of the effects of using different activation functions on the training of a digit classifier may be found on this website[6].

While the architecture of the network is constrained by the problem to solve, the accuracy of the solution is determined by the training set. This set contains both the inputs and their

[6] https://towardsdatascience.com/exploring-activation-functions-for-neural-networks-73498da59b02

corresponding outputs of real examples regarding to the task. The training set of an image classifier, for example, is arranged on the following way: the inputs, which are $2D$ arrays, are stored on a n-dimensional array, X on equation (7), where n is the number of images contained on the training set. The set contains another n-dimensional array, Y on equation (7), containing the *classes* of the images. The training of the network consists on looping the inputs through the network, calculating the residuals for the elements on the training set, and making small adjustments on the NN parameters, namely the weights and the biases, in order to get a smaller residual.

For this work, the chosen framework has been *Keras*[7], which is compatible with *Python*, and runs alongside the algebraic package *Tensorflow*. This framework is intuitive to work with, and several guidelines may be found as introduction on its website, as well as a complete installation guide. *Keras* supports GPU based *tensorflow* package, meaning that the networks can be trained in GPU instead of in CPU. In this way, the calculation speed is multiplied by at least 2 orders of magnitude, giving us the chance to explore deeper and more complex architectures for the networks. On our work, 2 GPU units have been tested and compared to a common CPU unit, in terms of the time cost when training deep complex networks. The CPU and one of the GPU units belong to a personal computer, and the second GPU is currently installed on one of the computing units in the Instituto de Astrofísica de Canarias.

3.1 Methods

Neural networks provide infinite possibilities regarding the customization of the system, in order to build a system that best satisfies our needs. In terms of learning ability, neural networks are characterized by their *parameters*, the *weights* and *bias* on equation (7). Simultaneously, these parameters are controlled by the *hyperparameters* of the NN, which include the number of hidden layers and the number of units on each layer. The parameters are usually determined following a process of trial and error. However, the choice of hyperparameters is a determining factor when defining the type of problem a network can solve. Such is the case of the *activation functions*, features that give chance to include non-linearity.

On this section we introduce a discussion regarding the possible choices for hyperparameters and architectures, as well as normalization choices for the dataset.

3.1.1 Architectures

We propose two different architectures to build a network capable of interpolating SSP models inside a 2-dimensional space. One of the approaches will be designed to learn the *weights* of the target spectra on the parameter space composed of the library parameters (e.g. age, metallicity). We will refer to this approach as A_1, which is essentially a traditional interpolator. The second approach will perform a direct fitting of the spectral range. The aim here will be for the network to learn the spectral features and how they vary in the parameter space. This approach will be called A_2. The tasks of both designs are represented in Figure 11.

[7] https://keras.io/

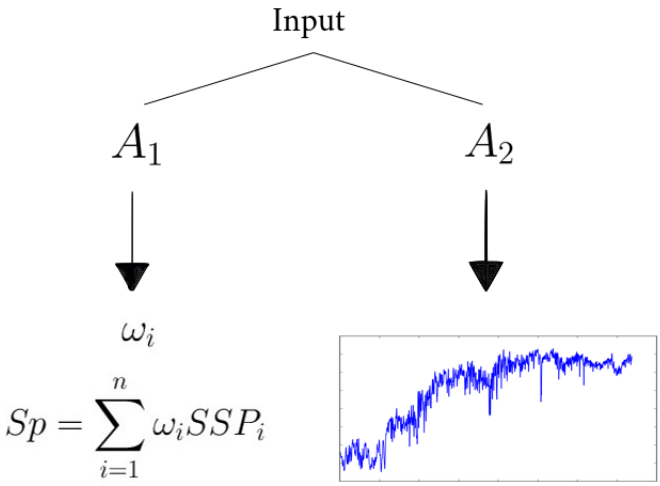


Figure 11: Representation of the propposed architectures. For a given set of input parameters, A_1 returns the weights of the corresponding model on the space defined by the SSP, while A_2 synthesizes the model.

In both cases, the input is an array containing all the parameter combinations corresponding to the templates. The output, however, will be different depending on the task.

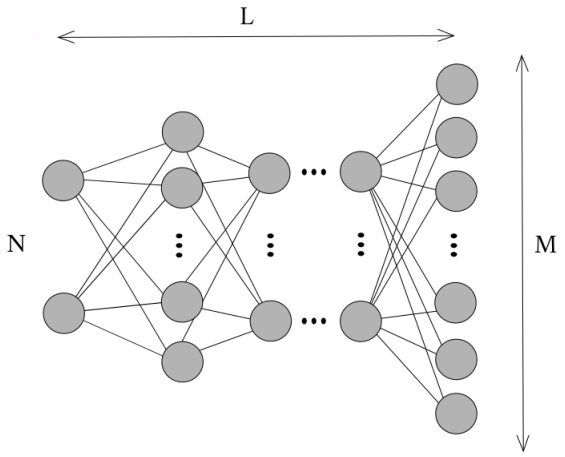


Figure 12: Scheme of a neural network suitable for both A_1 and A_2 architectures. L refers to the number of layers, and M to the number of neurons in the output layer. For both architectures, the input layer has 2 neurons for a $2D$ interpolation.

On Figure 12 we show a scheme of a generic architecture applicable for both A_1 and A_2 networks. The most important parameters on which the two cases differ are L, the number of total layers, and M, the number of neurons of the output layer. The first is a tunable parameter, but the last is restricted by the problem to solve. A_1 needs n neurons in the last layer, as pointed in Figure 11. On the other hand, A_2 requires a number of neurons equal to the λ channels that conform the SSP model.

3.1.2 Activation functions

Neural networks are often used to study non-linear phenomena, but the operations performed among the neurons of the hidden layers consist mainly on linear algebraic calculations. The hyperparameters that gives the option to introduce non-linearity are the *activation functions* of the layers. Different types of functions are required to solve different kinds of problems. On Figure 13 we show a selection of the most used ones.

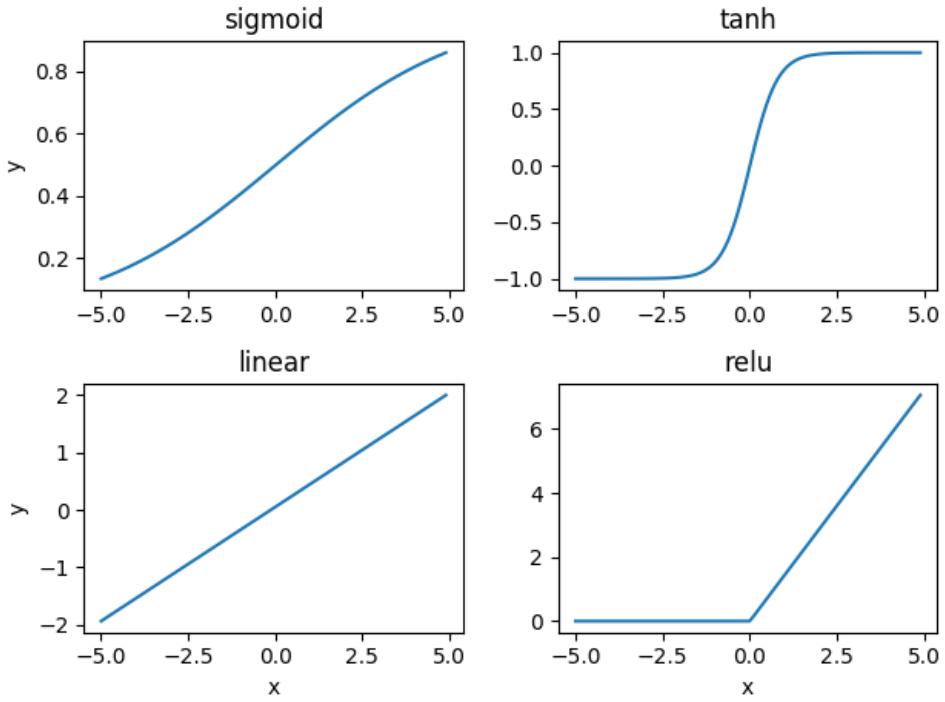


Figure 13: Selection of available activation functions. The functions *sigmoid*, *tanh* and *ReLU* are mainly used to introduce non-linearity, in contrast to the *linear* function. *ReLU* function is often used after convolution layers when classifying images, since no negative values are allowed for the pixel counts. Functions like *sigmoid* are often applied on the last layer of data classifying networks. The activation functions take an input X and produce an output Y whose range is restricted depending on the function.

All of the activation functions shown in Figure 13 are appliable on the hidden layers, but for both the interpolator and the spectral fitter, the last layer must have either a linear activation function, or no function at all. This is so because in regression problems the aim is to find an exact solution, not probabilities describing the similarity of an element with a given class, as it is in the case of classification problems.

The goodness of the solution is defined as how similar the predicted spectra is to the expected one along all the λ channels. Therefore, the parameter directly related to the precission of the network is the *loss* parameter, which turns out to be equal to the mean squared error. We consider that the network stops learning when the loss presents an asymptotical behaviour. In contrast with regression type problems, the performance of the networks designed for classification is represented by the parameter *accuracy*. Indpendently of the problem to solve, these parameters present an opposite trend as the training progresses: as the loss decreases the accuracy increases, given that the network is properly designed.

3.1.3 Normalization of the training set

When working with neural networks, the training set is ussually normalized according to the activation function of the output layer. For example, in classification networks, after a convolution layer comes a ReLU activation function. In this case, the normalization needs to rescale the training set so that no negative values are expected by the network. Afterwards, if necessary, the result of the NN may be transformed back to the original range of the dataset.

A possible option, applied on this work, is to perform an element-wise normalization, with the idea of creating a set of equally scaled elements. The SSP used for the training are scaled to 1 $M_{\odot}$, so the fluxes of young and metal poor SSP are much higher than the ones of old and metallic populations. Therefore, features of the latter group might be lost if the complete dataset is normalized at once. Dividing each SSP with its median we ensure a minimum feature loss caused by a global normalization.

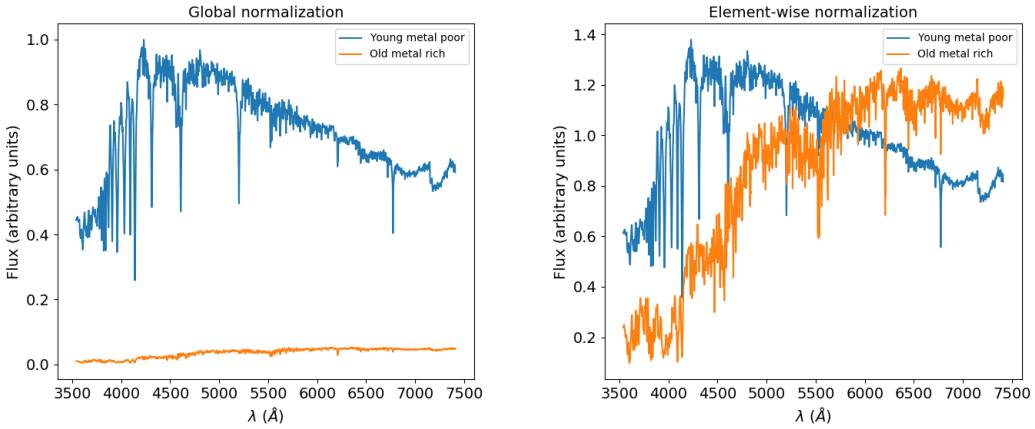


Figure 14: Examples of opposite elements of the training set after the normalization. On the left panel we show SPP after a global normalization between [0,1]. On the right panel we show the effect of the chosen normalization, element by element, applied to both SSP.

It is reasonable to assume that learning spectral features will be easier for the network if trained with elements shown on the bottom panel of figure 14. The details of the old metallic SSP are far better defined when an element-wise normalization is applied. Therefore, in the case of the global normalization, it is possible to suffer from loss of information.

3.1.4 Expanding to 3D and 4D

Making use of the architectures propposed above, it is possible to create networks that interpolate in an N-dimensional parameter space. The changes needed for this are minimal, since the task for both architectures remains the same. The difference lies on the fact that as input, the network will recieve N parameter values, instead on 2.

In order to achieve this, we make use of SSP models characterized by 4 physical parameters: age, metallicity, IMF slope value and abundance ratio [α/Fe]. From the libraries available on the MILES webpage, we chose 3 that were created combining BasTI isochrones with the MILES stellar library: one with solar scaled abundance ratio, a second with α enhanced abundance ratio, and a third whose models follow the abundance pattern of the Milky Way. Subsequently, we will name these libraries solar-scaled, α-enhanced and base libraries.

These 3 libraries contain more models than the one used on the PCA chapter, sorted in 53 ages, 12 metallicities and 14 IMF values, gathering a total of 8904 SSP each.

The transition from 2D to higher dimensions was studied by building and testing 5 different networks, each of them created to perform a different, more complex task.

Network1

This network incorporates the interpolator architecture, A_1, and will be trained with SSP of all ages and metallicities, for a single value of IMF slope. The SSP models belong to the base library.

Network2

The second one performs a 2D interpolation with an A_2 type network with SSP of all ages and metalicities, but again only one IMF slope value. The training set was the same as for Network1.

Network3

The third performs a full 3D interpolation on the parameter space defined by all the ages, metallicities and IMF slope values contained on the library containing the base models. This network incorporates architecture type A_2.

Network4

The fourth network performs interpolation in 3D, again with A_2, for a single value of IMF and 2 values of abundance ratio, alongside all the ages and metallicities. In this case, the training set combines SSP from the α-enhanced and solar scaled libraries, covering all the age-metallicity plane for a IMF slope value of 1.30.

Network5

The last one performs a full interpolation in the 4-dimensional parameter space, using A_2 type architecture. This case presents the largest training set, englobing a total of 8904 single stellar population models.

These five designs will help us to study how the inclusion of more parameters affects the learning of A_2 architecture. For example, Network1 and Network2 may be used to compare the performance of both arhitectures when trained with a more extense library. Network3 and Network4 extend the problem to 3 dimensions, and will help us understand the performance of the proposed architectures when facing more complex problems. Finally, Network5 will be the ultimate test for the architecture, increasing the complexity of the problem to a higher degree.

3.2 Results

In this section we present the results obtained regarding the tasks presented above. To begin with, we will discuss the performances of the training process for a given network in terms

of time cost, comparing the GPU devices and the CPU unit. Afterwards, we will present on the results regarding design choices and interpolation accuracies for all the cases formulated in the chapter.

3.2.1 Training times: CPU versus GPU

Focusing on architecture A_2, which requires the most complex architecture, a series of trainings have been performed using a CPU unit and 2 GPU units. This architecture results of most interest, since is the one that is actually trained to learn the features along the wavelength range in function of the spectral parameters.

Taking advantage of the computational power of the GPU units, in particular the P100, we performed a test designing a deep and complex layout for A_2. We trained the design with 78 SSP models, for a total of 1000 iterations.

On Table 3 we present the times taken on the training of the network shown in Figure 15 for all 3 devices.

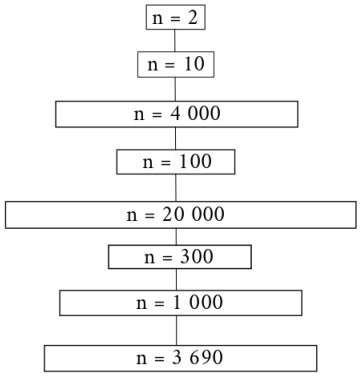


Figure 15: Architecture used for the training time comparison between CPU and GPUs. The network has a total of 8 layers, with varying number of neurons n on each.

Device	Time/step	Time
Intel i5-4200 (CPU)	1.01 s	578 m 16.86 s
GTX 765M (GPU)	3 ms	47 m 26.49 s
P100 (GPU)	$200 - 300$ μ s	137.455 s

Table 3: Times of the training for the spectral fitter: CPU versus GPUs. In all 3 cases the training was performed for 1000 iterations. The architecture of the network for these trainings is shown in Figure 15.

It is clear at first sight that the $P100$ GPU turns out to be the best choice when training deep complex networks. Compared to a normal $i5$ Intel CPU, the time cost is decreased for almost 3 orders of magnitude, which reduces the total time cost significantly.

We must point out that the accessibility to the P100 GPU unit has been of paramount importance when completing this work, since it provided opportunity of planning fast and long training sessions. However, the resources of the GPU unit were not exploited to the maximum, since the priority was to extend the dimensionality to 3 and 4 dimensions. For

further completion, the next step would consist in running extremely long trainings, in order to det the minimum residuals possible for the interpolations.

3.2.2 Architectures

Next, a $2D$ interpolation was performed on the plane defined by the parameters *age* and *metallicity* with both A_1 and A_2 architectures. As both are designed to produce the same outcome, the results are directly comparable. The training set used for this exercise contained 156 SSP, split in 26 ages and 6 metallicities.

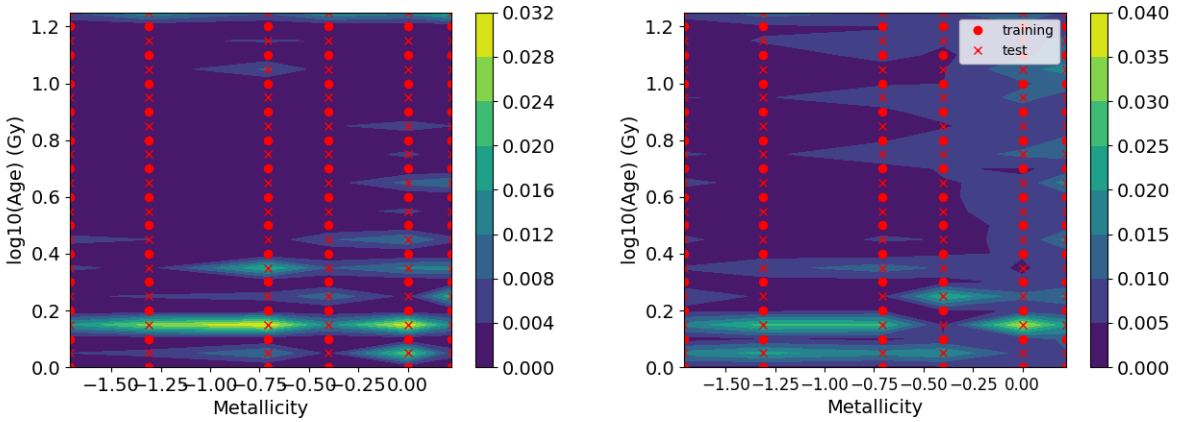
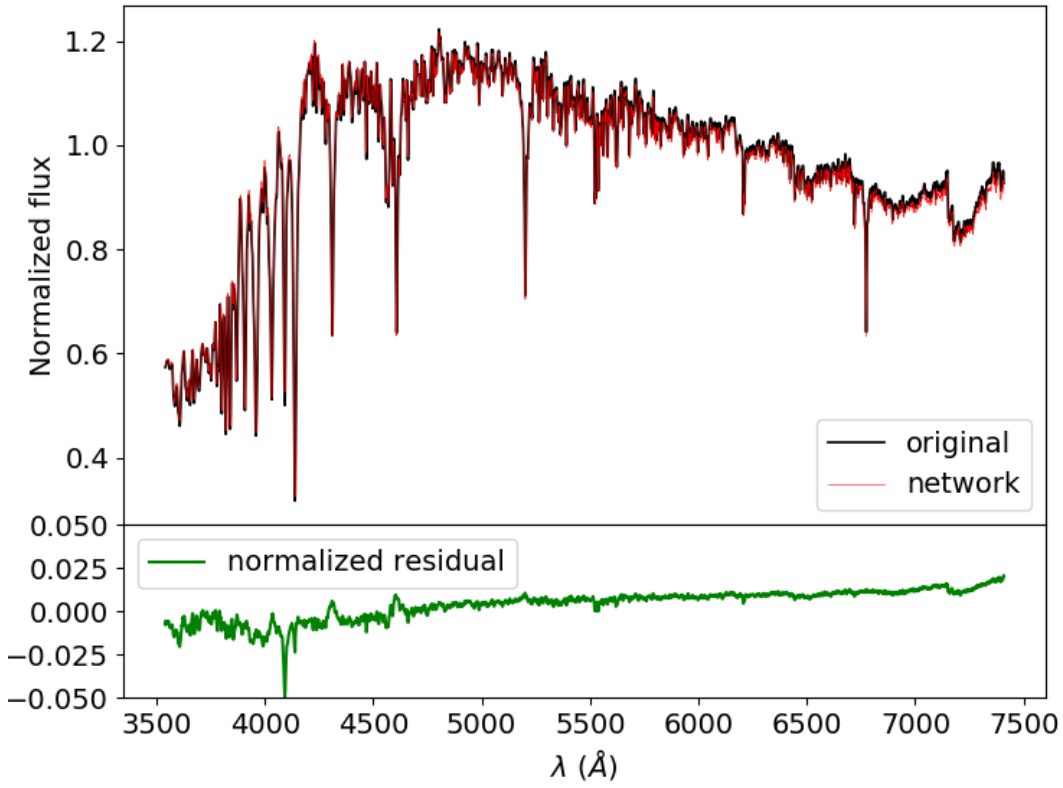


Figure 16: Comparison between the accuracies of architectures A_1 and A_2, when trained with 78 SSP. The interpolated SSP are represented with an X. The color code shows the standard deviation of the normalized residuals between the interpolated and their corresponding true SSP.

Looking at Figure 16 we can say that both architectures show to be capable of interpolating half of the training library with acceptable accuracy. The standard deviation of the normalized residuals is lower than 4 % in the case of the A_2 architecture, while for A_1 it peaks around 3,2 %. A_1 appears to interpolate better than A_2 on the central part of the domain. However, taking into account that the task of the latter architecture is much more complex to begin with, the results obtained are far from being negative.

The cases presented on Figure 16 correspond to networks designed to perform the same task. In order to prove the validity of the approaches, we requested both architectures to create a new SSP, corresponding to a pair of age and metallicity not contained either on the training nor the test sets. The results are plotted in Figure 17.



(a)

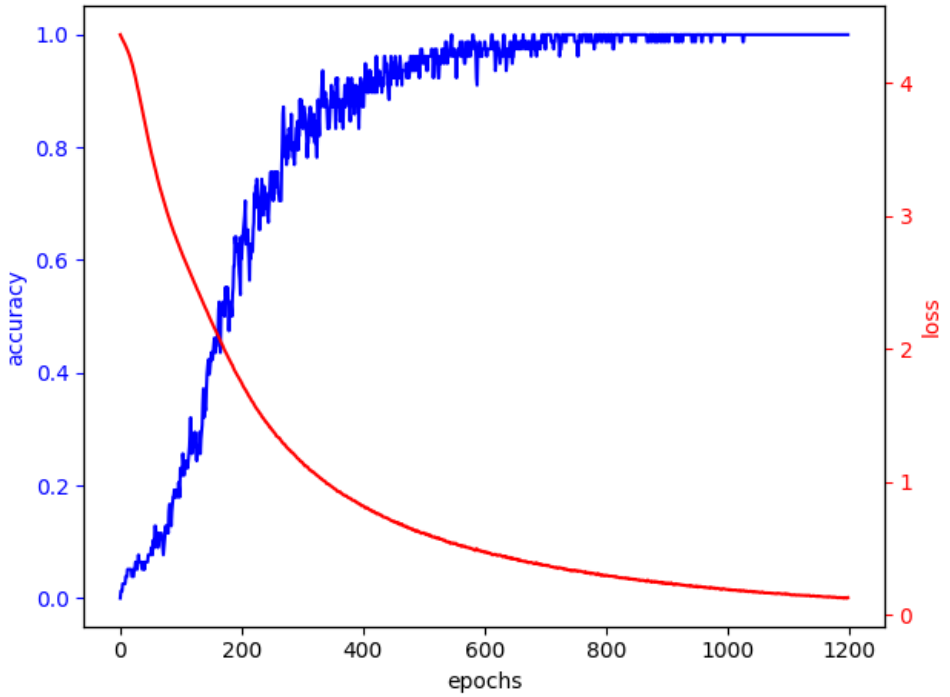
Figure 17: Comparison of the results of an interpolation for architectures A_1 and A_2. The pair of parameters corresponding to these SSP is $(logAge, metallicity) = (0.23, -0.80)$.

Figure 17 shows that very similar results are obtained with both A_1 and A_2 architectures. This is expected, since the same training set was used to train both models. Regarding the spectral distribution for the target pair of age-metallicity show prominent Balmer series and a medium spectral line density, as expected in an young and intermediate metallicity SSP.

3.2.3 Activation functions

The problems to solve by both of the proposed architectures are regression type problems. Thus, the final layer of the networks needed to have a linear activation function, or no function at all. We chose not to implement any activation function on the last layer for both A_1 and A_2. For the hidden layers, we selected the activation function ReLU, since we needed to have only positive values for the final result of the network.

On Figure 18 we present the the accuracy and loss of a training process of 1200 iterations, involving architecture A_1 and a training set containing 78 SSP. In this case, we included only one hidden layer with 100 neurons.



(a)

Figure 18: Parameters describing the training of the A_1 architecture. The red curve represents the loss, and the blue plots the accuracy. The training was performed for 1200 iterations.

3.2.4 Normalization of the training set

As pointed on the previous section, the normalization of the data might be a crucial decission when preparing the training set for a neural network. On Figure 19 we compare the efects that 2 different normalizations have on the output of the same architecture designed for the A_2 type network of Figure 12.

Looking at Figure 19 it appears that the element-wise normalization produces better results than a global normalization. As suspected, scaling all the SSP to the same range altogether leads to a loss of precission in certain areas of the 2-dimensional domain.

3.2.5 Expanding to 3D and 4D

Until this point, the study of neural networks was focused on finding architectures capable of interpolating on a 2-dimensional plane. Once tested the validity of such architectures, the next step consisted in the expansion into higher parameter dimensions. Network1, Network2 and Network3 were trained using the library with no considerations about the abundance ratio, holding the base models. On the other hand, Network4 and Network5 were trained combining the solar scaled and α enhanced libraries.

Network1

In Figure 20 we present the results obtained from Network1. We compare the goodness of the 2-dimensional interpolation after training A_2 architecture with 78 SSP, shown on the left panel,with the same exercise but trained with a larger set containing 318 models.

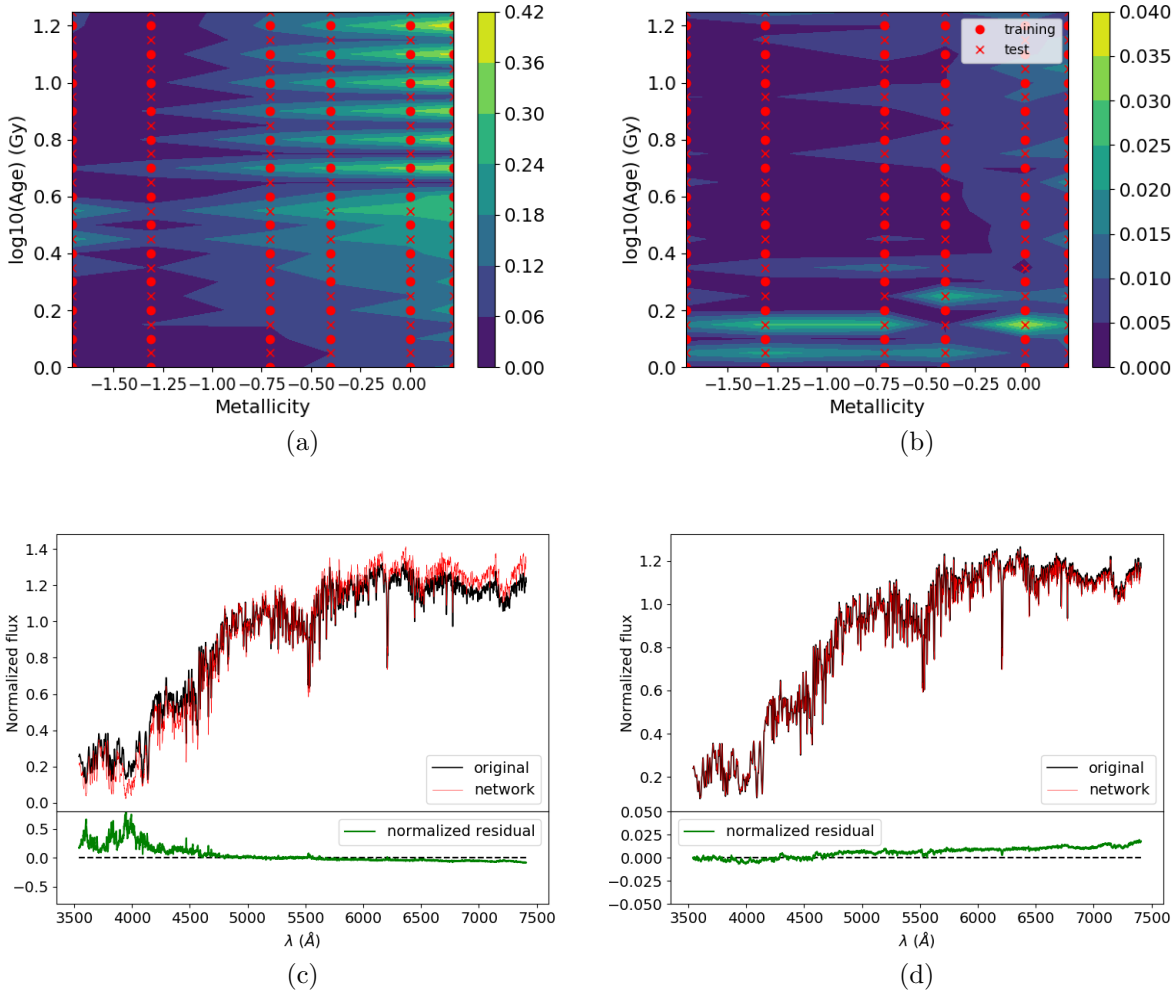


Figure 19: Comparison between the results obtained by 2 different data normalizations. The top left panel shows the performance of A_2 after normalizing the whole dataset between [0,1]. In the case of the top right panel, an element-wise normalization was applied, dividing each SSP with its median. On the bottom raw we present the corresponding interpolation for the same SSP: on the left panel applying a global normalization, and on the right panel with the more adecuate element-wise normalization. The upper limits on the colorbars show that for all the plane, the element-wise normalization is the best choice.

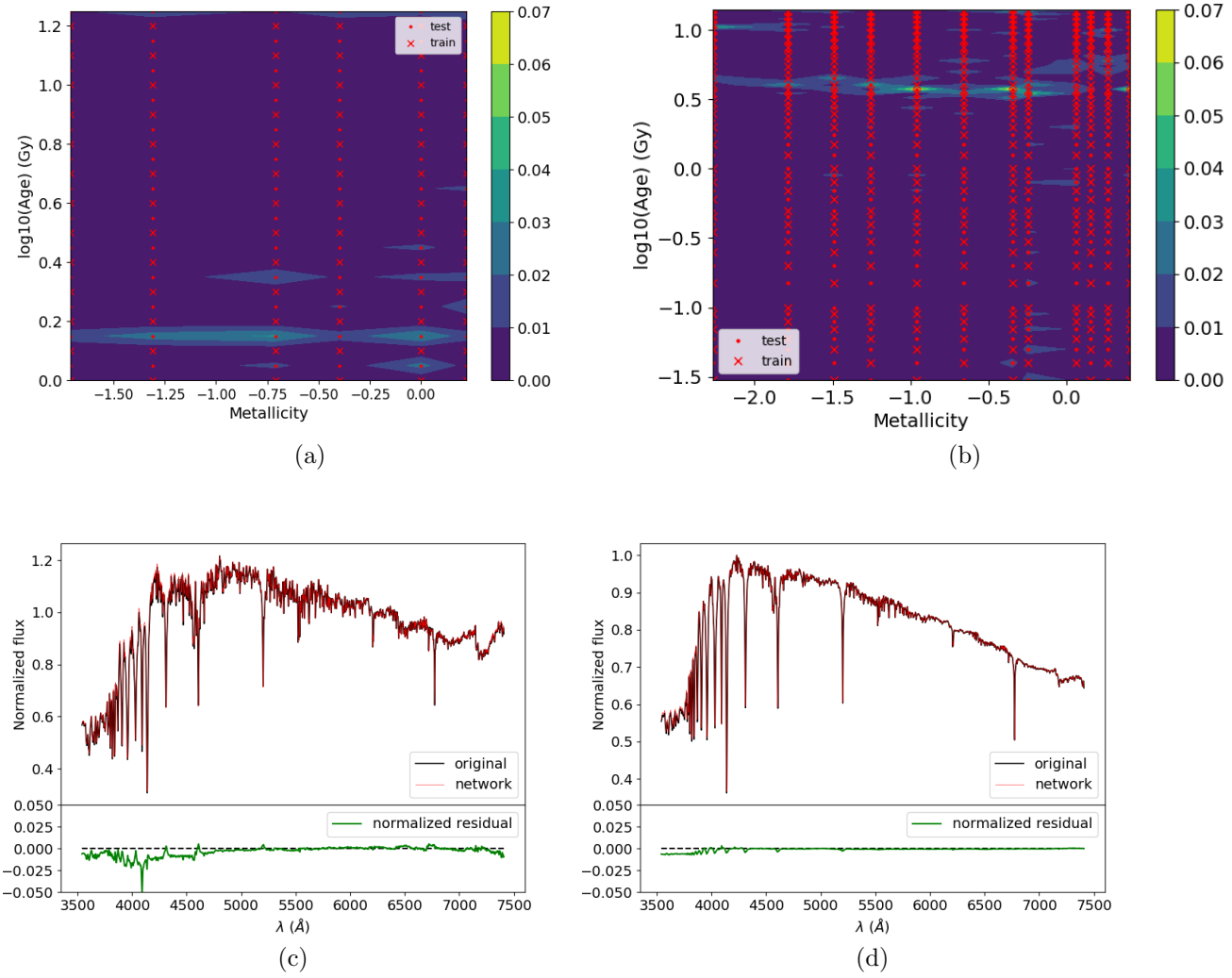


Figure 20: Comparison of the accuracy of architecture A_1 after increasing the size of the training set. On the top left panel we plot the results after training the network with 78 SSP, and on the bottom left panel we plot an example of the resulting interpolation. The top right panel shows the accuracy of the same network, after training it with 318 models, and the bottom left panel shows a sample corresponding the top right one.

For almost all the domain the standard deviation of the normalized residuals is comparable to when trained with a larger training set. For both cases presented in Figure 20 the training was performed for 2000 iterations.

Network2

When including al the IMF slope values of the library, the size of the training set increases significantly. Similar to the cases presented before, on figure **??** we plot the standard deviation of the residual of the interpolations on the plane defined by age and metallicity.

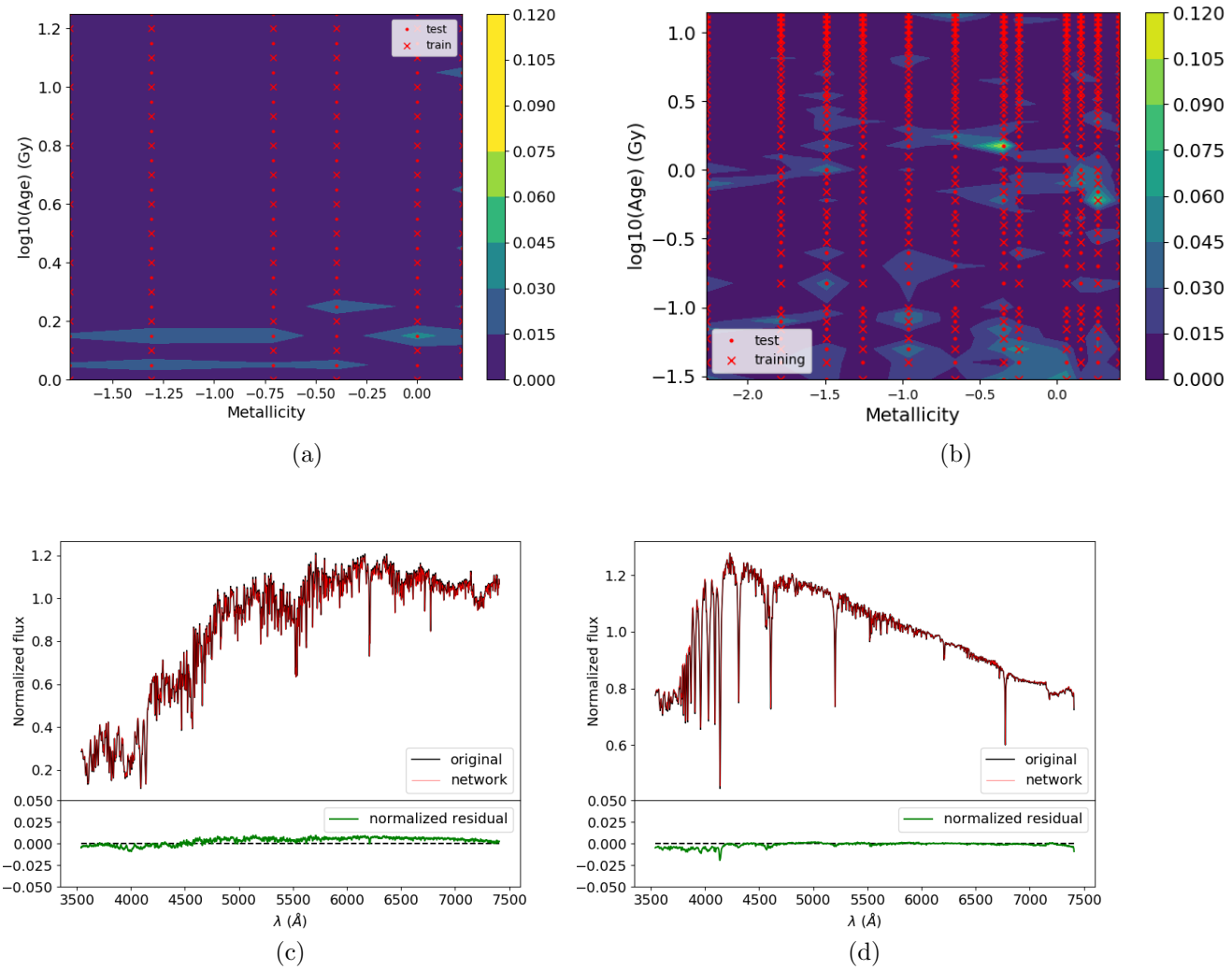


Figure 21: Comparison of the accuracy of architecture A_2 after increasing the size of the training set. On the top left panel we plot the results after training the network with 78 SSP, and on the bottom left panel we plot an example of the resulting interpolation. The top right panel shows the accuracy of the same network, after training it with 318 models, and the bottom left panel shows a sample for each case.

Network3

The third network included all the values for the IMF slope, constructing a complete 3-dimensional space. At this point, the complexity of the problem increased considerably, since tha training set got from having 636 elements, to a total of 8904 SSP models. We show the accuracy of 3-dimensional interpolation in Figure 22, representing the age-metallicity plane.

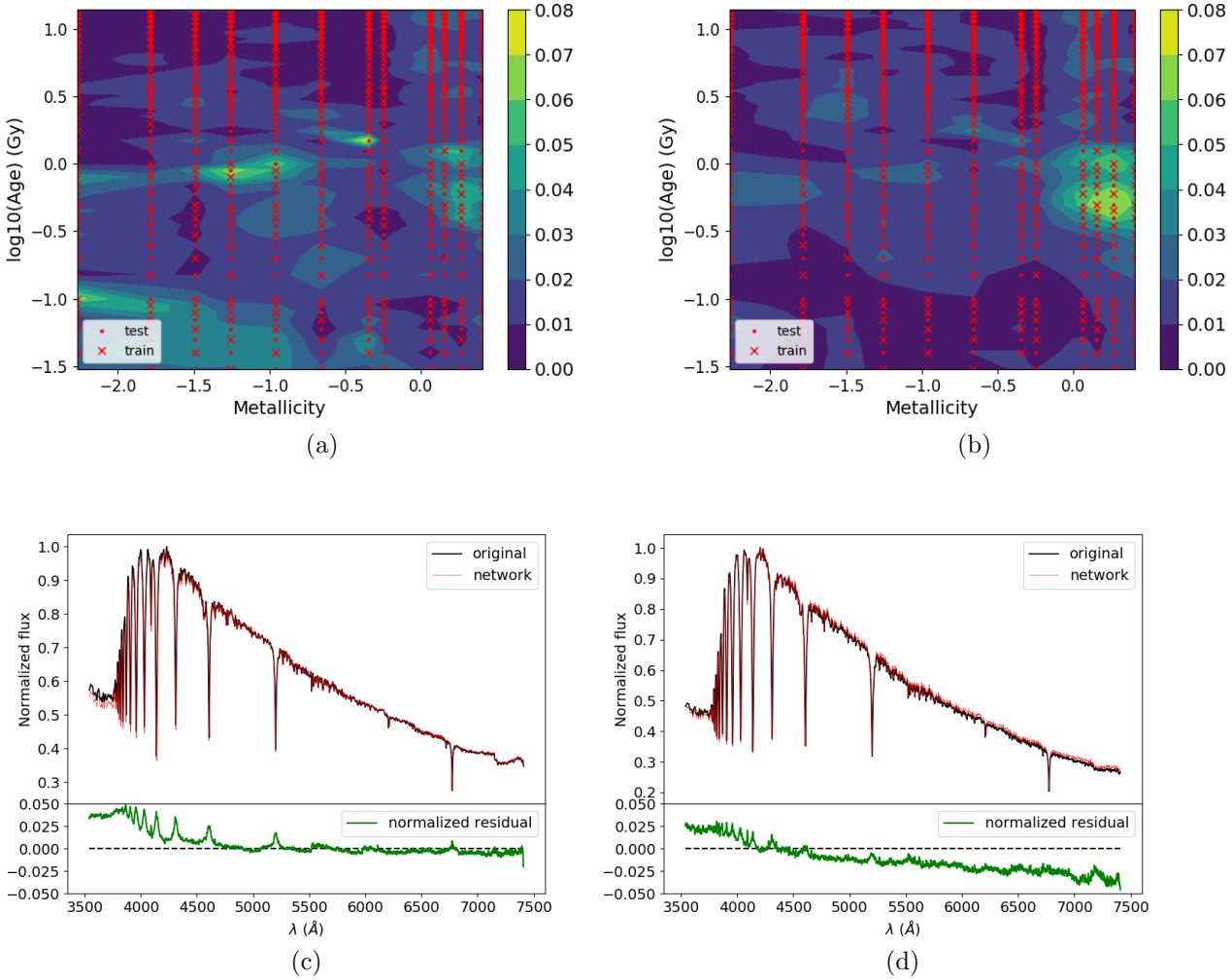


Figure 22: Comparison of the accuracy of architecture A_2 after including all 14 values of IMF. On the top left panel we plot the accuracy of the interpolation for the plane corresponding to a IMF value of 1.00. The top right panel shows mean of the accuracy across the IMF value axis. On the bottom panels we show 2 examples of interpolation in 3 dimensions.

Network4

Network4 was designed to include a 4^{th} parameter, the abundance ratio, and see how the accuracy of the network behaves when inserting IMF slope and abundance ratio separately.

The training set was created taking SSP of all ages and metallicities for a single IMF value from the libraries with α enhanced and solar scaled abundance ratios. In Figure 23 we present the results of the interpolation on the age-metallicity plane for both cases of the abundance ratio, with a IMF slope value of 1.30.

The results plotted in figure 23, show that, for a single value of IMF, the results obtained after introducing a 4^{th} parameter are comparable to the previous cases. Therefore, we can assume that the architecture of the spectral fitter is appropriate for interpolating spectra in a multidimensional parameter space.

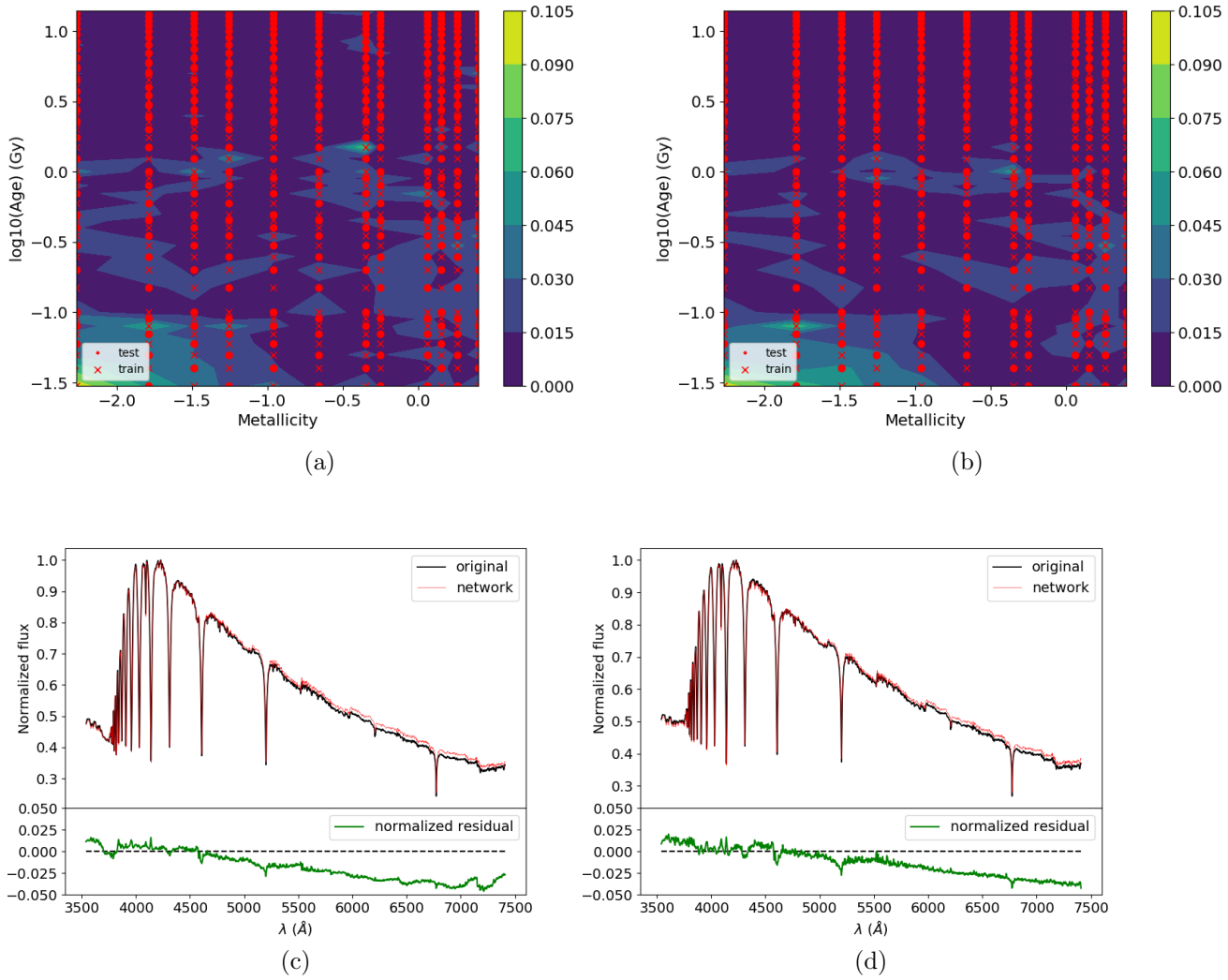


Figure 23: Accuracy of the interpolations after including abundance rationas parameter. A single IMF value of 1.30 was chosen, while maintaining the age-metallicity grid. The top panel left corresponds to a solar scaled abundance ratio, and the top right panel to the α enhanced library. On the bottom panels we show the interpolation of a SSP of the same age, metallicity and IMF slope value, with a solar scaled abundance ratio on the left, and α enhanced abundance on the right.

Network5

After adding the remaining IMF values, the training set for the fourth network increased to a total number of 17808 SSP. Again, half of them were used for training, keeping the other half for testing. Having such a large training set increases considerably the time taken on the training. An iteration of the whole set along the network takes about $11s$ when training on the P100 GPU. Compared to the times presented on table 3, the effects of the size of the training set are noticeable. However, the time cost increase is compensated by the accuracy of the 4-dimensional spectral fitter, as it can be seen in figure 24.

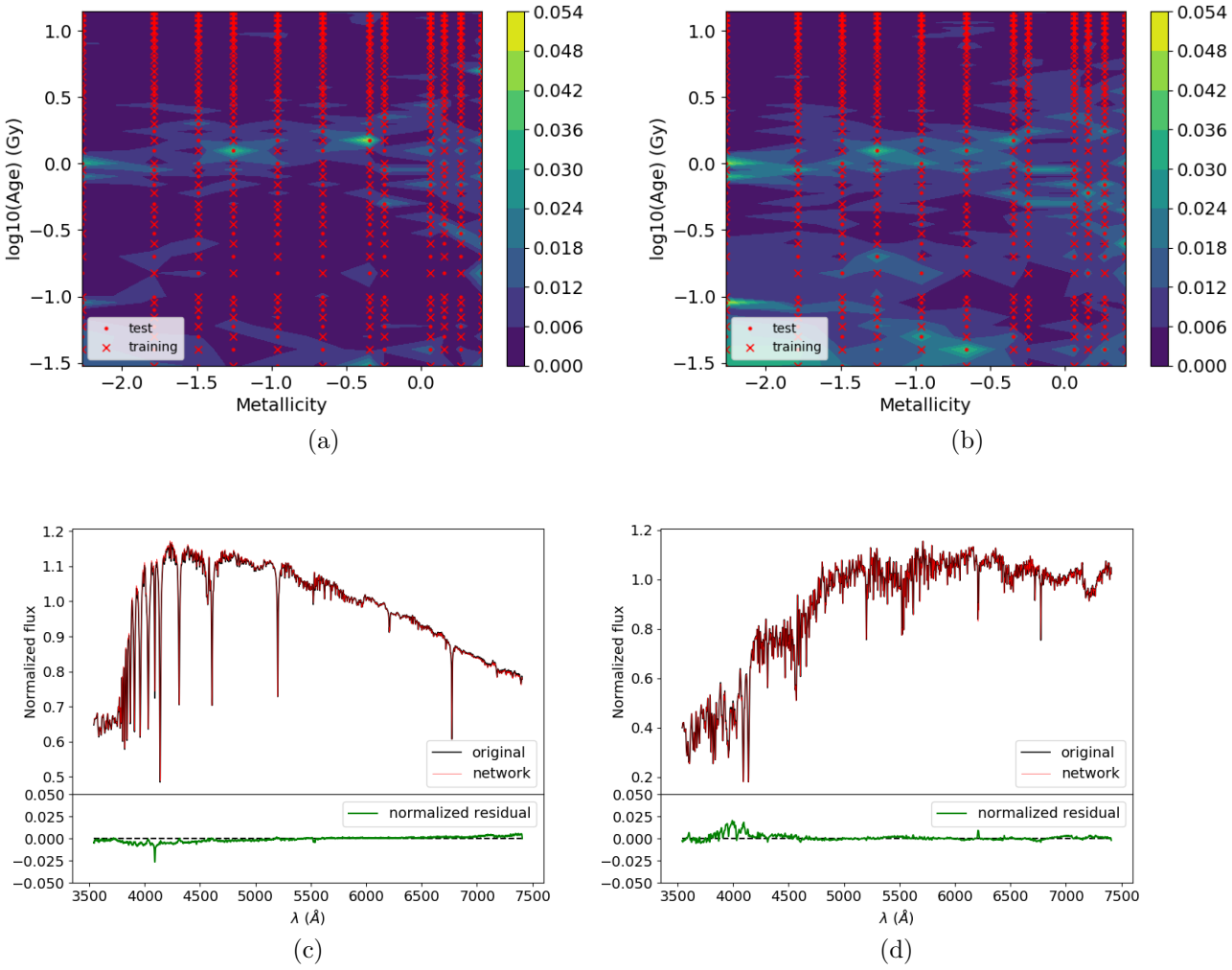


Figure 24: Accuracy of the learning process after including abundance ratio as parameter and all of the 14 IMF values. The top left panel shows the mean across the IMF axis for the solar scaled abundance ratio; the top right panel shows the equivalent for the α enhanced case. On the bottom panels we plot the results of two interpolations: on the left panel we show a young metal poor SSP, and on the right panel an old metal rich SSP, performed with the case trained with α-enhanced library.

Looking at the progression of the residuals, ilustrated along this subsection, we note that the behaviour of the spectral fitter improves as the size of the training set increases. The idea results quite intuitive, since for a traditional interpolation better results are obtained in a dense populated parameter space. The improvement of the accuracy on the age-metallicity plane as we moved from 2, to 3 and finally to 4 dimensions also corroborates the idea that large training sets result in better trained neural networks.

However, this comes with its cost, since the trainings of the 3 and 4-dimensional A_2 networks take far much more time than the 2-dimensional case. For the latter, a single iteration around 78 SSP took between $200 - 300\mu$ s, training on the P100. The largest of the training sets, the one corresponding to the full $4D$ spectral fitter, and whose results are presented in figure 24, took 11 seconds iterating once along 8904 SSP. Nonetheless, as the

training needs to be carried out only once, the time considerations are negligible when using the network.

4 Conclusions

On this last section we present the conclusions to which we arrived after completing the tasks proposed through this work. To begin with, we will discuss the implementation of the PCA technique as dimensionality reduction method, focusing on how the information of the subject dataset is preserved when performing the principal component analysis. Afterwards, we will focus on the implementation of Machine Learning techniques with the objective of training neural networks capable of interpolating SSP models in an multidimensional parameter space.

PCA resulted to be an effective method when reducing the dimensionality of libraries comformed by SSP models. The accuracy of the reconstruction of the PADOVA library, shown in figure 3 indicates that the principal components hold the essential information encoded in the original library. Each of the SSP models was reconstructed as a lineal combination of principal components, and the standard deviation of the residuals obtained after the comparison are below 2% on all the parameter space defined by the library. The principal components resulting from the PCA were used afterwards to recover the kinematics and populations of several galaxies, by means of the pPXF routine. For this exercise, we calculated these parameters using both the SSP library and subset containing 10 PC. The results obtained for the kinematic parameters v, σ, h_3 and h_4, though not exactly the same, showed to converge on similar values as the expected results from the SSP library, as it can be noted by looking at figure 8. The time cost has been drastically reduced, since the pPXF code run using 10 principal components instead of 156 SSP. On the table below we present a summary of the time performances regarding the study of the 15 galaxies chosen for the kinematic analysis.

For the second part of the work, we developed a method of interpolating the single stellar population models used by pPXF on a multidimensional parameter space. In order to achieve that, we studied the basics of neural network design, approaching the problem from two different ponts of view. On one hand, we designed and tested an architecture we called A_1 whose task was to perform a traditional interpolation, calculating the projection of a target SSP on among the preexisting ones. The second architecture was designed to return the target SSP directly. This last case results of most interest, since the network, if trained properly, learns the spectral features corresponding to a combination of physical parameters, and we named it A_2.

As shown in section 3.2, both of the proposed architectures proved to be capable of interpolating SSP in a 2-dimensional parameter space formed by the age and metallicity of the models. The networks were trained with half of the library used for the PCA analysis of section 2, and tested with the remaining half, so that exact models were available to compare with the output of the networks. Both architectures showed to be able to replicate the test group with an uncertainity below 4%. During the development of the networks, several aspects regarding the design were considered, as not all networks are suitable to perform a given task. In particular, we found that the data normalization has a significant impact on the output of a network. This was tested with architecture A_1, since the other case already had a normalized training set.

Heading to the final goal, we extended the architecture A_2 into 3 and 4 parameter dimensions, making use of libraries created using BasTI isochrones, instead of PADOVA isochrones. These libraries, characterized by 4 spectral parameters each, gave us chance of creating training sets that covered a higher dimensionality. In this case, similar results were found in comparison with the netwoks trained with the smaller library. The performance of the A_2 design improved for almost all the 2-dimensional plane as the training set increased in size.

This is noticeable in figure 22, but more so in figure 24, where the maximum uncertainity is below 5.4% while the complexity of the problem had increased considerably. This seems to be caused by the increase in size of the training sets, when interpolating in 3 and 4 dimensions.

Looking at the results presented along the third section, we are able to confirm that it is possible to design artificial neural networks capable of synthesize SSP models parametrized by spectral parameters. Using relatively simple architectures, we have created designs capable of interpolating SSP instantly in a 4-dimensional parameter space, which, as methods like PCA, might result in a decrease of computational time cost.

As an extension, for a future work, it would be feasable to design architectures that, alongside with population parameters, would be able to convolve SSP models with kinematic parameters, such as v and σ, thus creating neural networks that would be capable of generating models based on populations and kinematics.

References

A Vazdekis, P Sánchez-Blázquez, J Falcón-Barroso, AJ Cenarro, MA Beasley, N Cardiel, J Gorgas, and RF Peletier. Evolutionary stellar population synthesis with miles–i. the base models and a new line index system. *Monthly Notices of the Royal Astronomical Society*, 404(4):1639–1671, 2010.

Honglin Lu, Hongyan Zhou, Junxian Wang, Tinggui Wang, Xiaobo Dong, Zhenquan Zhuang, and Cheng Li. Ensemble learning for independent component analysis of normal galaxy spectra. *The Astronomical Journal*, 131(2):790, 2006.

Yan-Mei Chen, Guinevere Kauffmann, Christy A Tremonti, Simon White, Timothy M Heckman, Katarina Kovač, Kevin Bundy, John Chisholm, Claudia Maraston, Donald P Schneider, et al. Evolution of the most massive galaxies to z= 0.6–i. a new method for physical parameter estimation. *Monthly Notices of the Royal Astronomical Society*, 421(1):314–332, 2012.

Shai Ronen, Alfonso Aragón-Salamanca, and Ofer Lahav. Principal component analysis of synthetic galaxy spectra. *Monthly Notices of the Royal Astronomical Society*, 303(2): 284–296, 1999.

J Sánchez Almeida, J Alfonso L Aguerri, Casiana Munoz-Tunón, and Angel De Vicente. Automatic unsupervised classification of all sloan digital sky survey data release 7 galaxy spectra. *The Astrophysical Journal*, 714(1):487, 2010.

Waseem Rawat and Zenghui Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation*, 29(9):2352–2449, 2017.

Dinesh P Mital and Leonard Chin. Application of neural networks in robotic control. In *TENCON'98. 1998 IEEE Region 10 International Conference on Global Connectivity in Energy, Computer, Communication and Control*, volume 1, pages 162–165. IEEE, 1998.

Christopher M Bishop and CM Roach. Fast curve fitting using neural networks. *Review of scientific instruments*, 63(10):4450–4456, 1992.

A Asensio Ramos, IS Requerey, and N Vitas. Deepvel: Deep learning for the estimation of horizontal velocities at the solar surface. *Astronomy & Astrophysics*, 604:A11, 2017.

Pavel Hála. Spectral classification using convolutional neural networks. *arXiv preprint arXiv:1412.8341*, 2014.

Eric Emsellem, Michele Cappellari, Reynier F Peletier, Richard M McDermid, R Bacon, M Bureau, Y Copin, Roger L Davies, Davor Krajnović, Harald Kuntschner, et al. The sauron project–iii. integral-field absorption-line kinematics of 48 elliptical and lenticular galaxies. *Monthly Notices of the Royal Astronomical Society*, 352(3):721–743, 2004.

Michele Cappellari and Eric Emsellem. 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, 2004.