Facultad de Ciencias
Universidad de La Laguna

PHYSICS SECTION
FINAL DEGREE PROJECT

MACHINE LEARNING AND DEEP LEARNING IN SOLAR PHYSICS

Yazmina Zurita Martel

Supervised by
Dr. Andrés Asensio Ramos

July 2020

Contents

1 Abstract

There exists a growing interest in the application of Machine Learning methods to Astrophysics, as they have proven to be very useful in the past. Following this line of work and motivated by a reduction in computation time, we explored the use of an artificial neural network (ANN) in the synthesis of Stokes profiles through the direct mapping of solar atmosphere magnitudes and profiles. First, some basic concepts regarding Machine Learning and the problem at hand are presented, followed by a complete description of the ANN training process. Finally, a Stokes profile inversion (which requires a great number of synthesis) is carried out using an optimization algorithm to recover the solar atmosphere magnitudes.

Resumen

Existe un interés creciente en la aplicación de métodos de Aprendizaje automático a Astrofísica, ya que han demostrado ser de gran utilidad a lo largo de los años. Siguiendo esta línea de trabajo y motivado por una reducción en el tiempo de computación, se explora el uso de una red neuronal artificial (ANN) en la síntesis de perfiles de Stokes mediante un mapeo directo de magnitudes de la atmósfera solar y perfiles. Primero, se exponen una serie de conceptos básicos sobre Aprendizaje automático y el problema que nos ocupa, seguido de una descripción completa del proceso de entrenamiento de la ANN. Por último, se realiza la inversión de un perfil de Stokes (lo cual requiere un gran número de síntesis) haciendo uso de un algoritmo de optimización con el fin de recuperar las magnitudes de la atmósfera solar.

2 Introduction

En primer lugar, se expone la motivación del trabajo junto a la razón por la cual los métodos de Aprendizaje automático (ML) han ganado popularidad en Astrofísica, además de enumerar ejemplos concretos de su uso. Se prosigue con una presentación detallada de conceptos claves en ML, tales como red neuronal artificial, parámetros del modelo, retropropagación o entrenamiento supervisado. En un último sub-apartado, se definen los parámetros de Stokes y los métodos de síntesis e inversión, que realacionan los perfiles con una serie de magnitudes de la atmósfera solar mediante la ecuación de transporte radiativo.

2.1 Motivation

Machine Learning algorithms have been gaining popularity throughout the years among a great number of sectors such as Finances, Logistics, medical diagnosis or transport, offering different and better approaches, solving challenging problems and opening the door to brand-new technology.

These methods have experienced a rise in Astrophysics too [Longo et al., 2019]. Datasets are quickly becoming larger and more complex, therefore the need for a tool able to process high-dimensional data quickly. In this regard, Machine Learning contributes to anomaly and object detection, classification, clustering, as well as inference, prediction and faster problem resolution.

2.2 Machine Learning and Deep Learning

Machine Learning, a subfield of Computer Science and a branch of Artificial Intelligence, develop techniques to give computers the ability to "learn" (i.e., progressively improve performance on a specific task) with data, without being explicitly programmed.

At the heart of every machine learning algorithm lies a model, a set of successive operations needed to transform raw data (input) into some desired result (output). In this project, artificial neural networks (ANNs) will be used as models. ANNs employ nodes or neurons that are linked to one another in a specific way, conforming the architecture. Each connection has a weight associated, a value representing a neuron influence over another. These model parameters are unknown at first, and the objective is to iteratively adapt them through a training process until a good enough performance is achieved.

Deep Learning refers to machine learning algorithms whose architecture allows learning in multiple successive levels of abstraction, establishing a hierarchy among the elements found in data so that complex high-level features are defined in terms of simple lower-level ones. This technique usually employs ANNs where neurons are arranged in layers and information only moves forward from the input to the output. Taking the picture of a bird species as an example, the first layers would excel detecting lines; the next group, borders, and so on with feathers, body parts, birds in general and, finally, concrete species of birds. This behaviour can be visualized in [Zeiler and Fergus, 2013]. For groups of layers to be able to specialize in such a way, a large number of them is required, being this the main difference with other machine learning algorithms using ANNs.

The fundamental process involving linear neurons is displayed in Figure 1. Note that this is just a general example and variations are applicable. The first step is taking the values stored

in previous neurons and multiply by the corresponding weights. Then, a bias is added, a neuron unconnected to previous ones that adds a constant term to the operation. Next, a weighted sum of these terms is calculated and fed to an activation function, which normally is used to introduce non-linearity and help the ANN learn complex data. There could be other features attributed to a particular activation function (e.g. restricting values within a range). Finally, the result of these operations is transmitted to the next group of neurons.

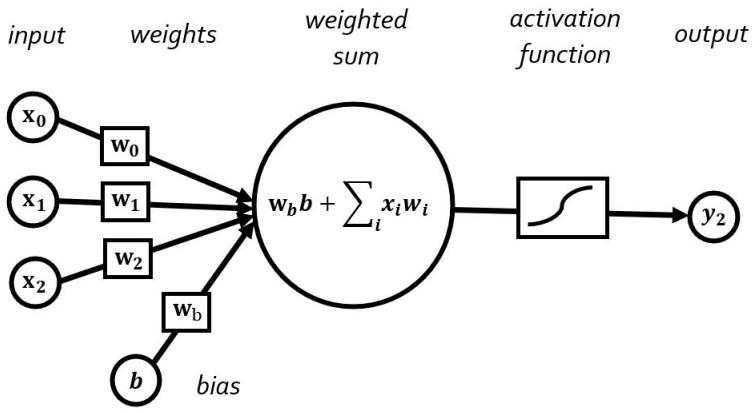


Figure 1: Main components of ANNs with linear nodes and the basic interactions that take place whitin them.

The process outlined above is applied to every neuron composing the network until a final group of nodes is reached which returns the model output.
In supervised learning, every input is associated with an expected output provided by the user. This paired information is known as "labeled data". Here, the output predicted by the model is compared against the expected outcome via a loss function.

Weights must be adapted to minimize the loss function and improve the model. A popular and fast method to do so consists in, first, applying the so-called backpropagation algorithm, which computes the gradient of the loss function with respect to any weights or bias in the network for every iteration (knows as epoch) and, second, couple it with another algorithm that update the weights taking steps proportional to the negative gradient calculated previously. The step size is known as the learning rate.

2.3 Stokes profiles and inversions

The outer region of the Sun is known as the solar atmosphere and can be divided into three different layers: the photosphere, the chromosphere and the corona.

Light stems from the photosphere and escapes the Sun after passing through the higher layers, and once this radiation is observed, it can be characterized by its spectrum, intensity and polarization state. There are several possible descriptions of radiation, but a convenient one is the given by the Stokes parameters (I, Q, U, V) [Stokes, 1852, Chandrasekhar, 1960]. These quantities

correspond to a sum or difference of easily measurable intensities when light is viewed through a polarizing filter oriented at various angles. I represents the intensity, while Q, U and V account for the linear polarization, inclination of the plane of polarization, and circular polarization.

Once these parameters are obtained for a certain range of wavelengths (i.e. the Stokes profiles are acquired), one may ask what are the physical conditions of the solar atmosphere that originated them and how to infer these magnitudes.

Nowadays, the combination of synthesis and inversion methods constitutes a powerful technique that yields good estimates. In synthesis, Stokes profiles are generated from sets of atmospheric quantities (models) employing the radiative transfer equation (RTE). Taking into account the polarization effect that the scattering of photons and the solar magnetic field have over radiation, the RTE takes the form

$$\frac{d\mathbf{I}}{d\tau_c} = \mathbf{K}(\mathbf{I} - \mathbf{S})^1$$

where $\mathbf{I} = (I, Q, U, V)$ is the Stokes vector, τ_c the optical depth at the continuum wavelength, $\mathbf{S}$ the source function vector and $\mathbf{K}$ the propagation matrix. From this equation, it can be drawn that the changes in the Stokes vector with respect to the optical depth are due to radiation absorption and dispersion on one hand, and photon emission on the other, represented by **KI** and **KS** respectively.

To execute a synthesis, one must provide temperature T, total preassure P, line-of-sight velocity V_{LOS}, microturbulent velocity V_{mic} and magnetic field strength $|\bar{B}|$, inclination γ and azimuth θ.

An inversion takes place when the state of the atmosphere is inferred from observed Stokes profiles. This can be done by iteratively synthesizing Stokes profiles and comparing them with the observation, carrying out an optimization process over the physical conditions, with the objective of minimizing the difference between the real and synthesized profile until a defined threshold is reached. The magnitudes from which the profile with the greatest agreement was synthesized, will constitute the best estimates of the atmosphere state that generated the observed profile.

3 Objectives

Se presenta el objetivo de este trabajo, partiendo del hecho de que realizar inversiones usando la ecuación de transporte radiativo para luz polarizada es un proceso largo y costoso computacionalmente. Se identifica entonces la ANN como una posible alternativa en el cálculo de síntesis.

The RTE for polarized light describes a very involved, non-local, non-linear problem, known as non-local thermodynamic equilibrium (NLTE). Being a vector differential equation, the presented RTE can be considered as a set of four coupled differential equations whose resolution requires the calculation of many parameters, progressively enlarging computation time and resources.

[1] [del Toro Iniesta, 2003, p. 116]

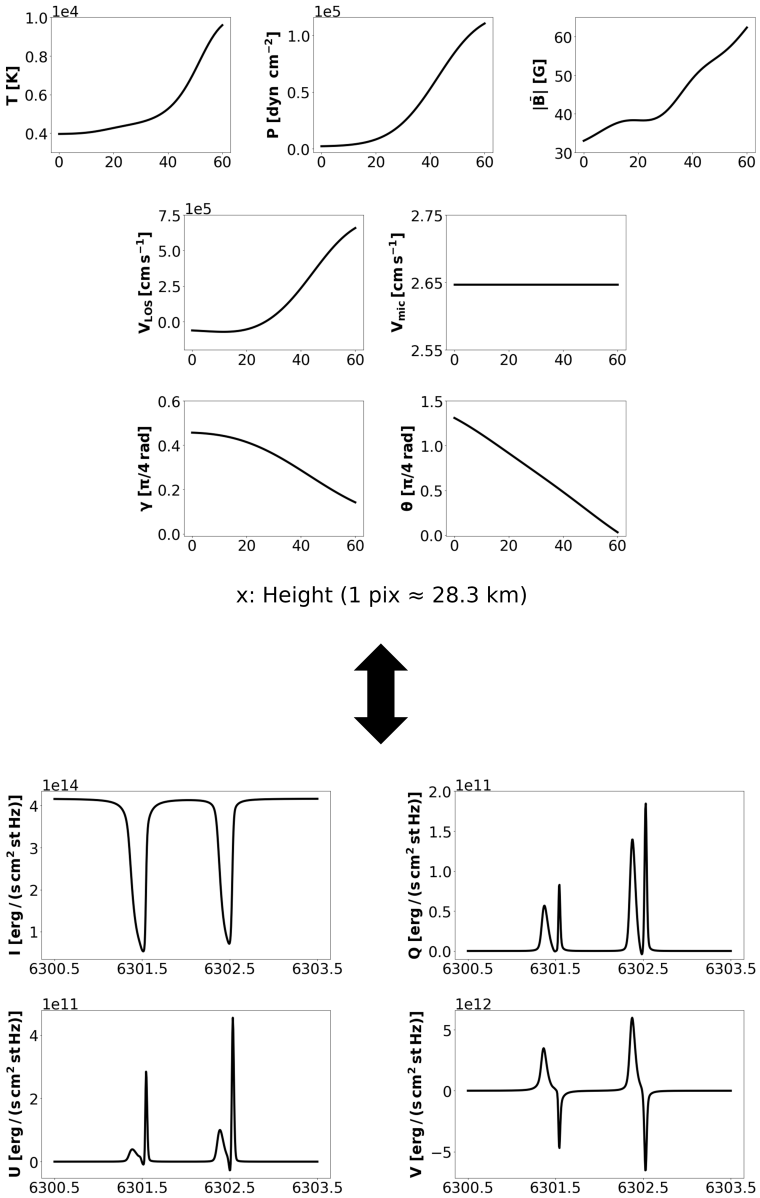


Figure 2: Above, set of magnitudes that conform the atmosphere model. Below, the Stokes profiles corresponding to two lines of Fe I. As has been discussed, one can be obtained from the other making use of synthesis and inversion methods.

This procedure starts with the calculation of the electronic and gas pressure by means of a differential equation, as hydrostatic equilibrium is typically assumed. Next, atomic level populations must be calculated. In the local thermodynamic equilibrium approximation, these are described by an analytical expression, but its calculation in NLTE is much more complicated and the inclusion of iterative operations is needed. Once the populations are known, the propagation matrix and the source function vector can be computed, which involves the evaluation of Voigt functions for line profiles for numerous wavelengths and heights. Note that inversions demand the synthesis of numerous Stokes profiles until an optimal approximation is reached, following the just described procedure every time.

This is what motivates a different approach to transfer modelling that proves to be fast and computationally efficient. Under this frame, ANNs represent a promising alternative to find an approximate numerical representation of the polarized radiative transfer calculations. Once the ANN is trained and the weights fixed, it is only a matter of introducing the atmospheric quantities as inputs in the model and, after some simple operations, obtain the corresponding Stokes profile.

Then, the objective of this project is to build an artificial neural network and study its effectiveness in the synthesis and inversion of Stokes profiles, evaluating how good of an alternative it is to radiation transfer modelling.

4 Methodology

La metodología de estudio se desarrolla a continuación. Se señalan las características del set de datos que se usará para entrenar la red y se comenta el tratamiento previo al que se debe someter, el cual involucra su degradación, análisis en componentes principales y escalado. Seguidamente, se expone la configuración específica de la ANN y del entrenamiento usado, mostrando, entre otros, la arquitectura de la red, la función de activación y de error y el algoritmo de retropropagación escogido, justificando en cada caso la razón de su elección. Por último, se señala brevemente el procedimiento seguido para realizar la inversión.

4.1 Data provided

The atmosphere simulation that will be used in this work is conformed by 288 x 288 pixels images of its surface extending 61 pixels in height. This cube is computed for every magnitude needed in the synthesis (section 2.3), as it is shown in Figure 3 for temperature, where the darker side corresponds to internal layers while the brighter one face to the exterior of the Sun.

The set of magnitudes shown in Figure 2 is extracted fixing a pixel in the square images and moving along the height axis. Making use of available codes [Socas-Navarro et al., 2015], a Stokes profile can be generated from each one of these sets, 288 x 288 in total, producing the labelled data that will be used to train the ANN. We will focus on two Fe I lines located at 6301.5 and 6302.5 Å in a range of 601 wavelengths.

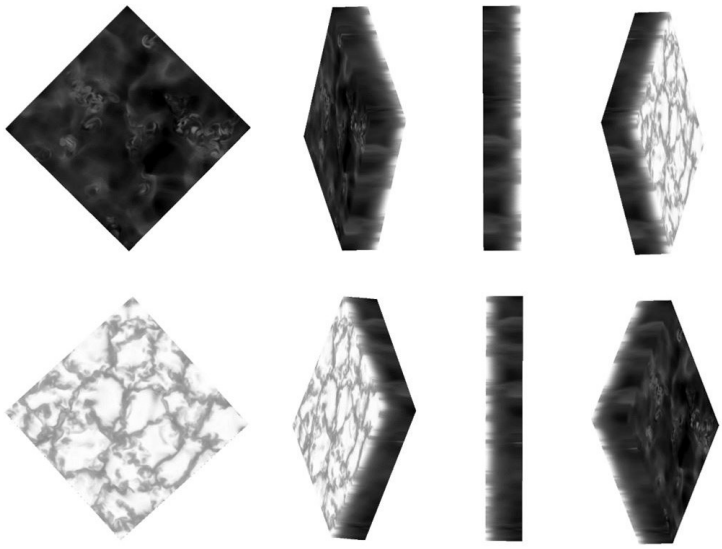


Figure 3: A 288 x 288 x 61 pixels cube of temperature rotating to the left arounf a vertical axis. The transition from cooler, lower layers to hotter, higher ones can be appreciated, along with the convection cells. Repository: napari contributors (2019). napari: a multi-dimensional image viewer for python. doi:10.5281/zenodo.3555620

4.2 Data preparation

4.2.1 Degradation of the synthesis

The Stokes profiles generated are closer to a theoretical ideal than they are to real observations taken by a telescope. To resemble the latter, it is necessary to degrade the former, which can be achieved through a convolution with the point spread function (PSF).

The PSF describes the spatial distribution of intensity in the focal plane of an imaging system when it is illuminated by a point source. Optical instruments like telescopes are affected by the diffraction of light and aberration effects that yield blurred spots instead of perfect points. This response to small sources of light as stars or quasars is contained in the PSF.

The image recorded by an imperfect device S_{deg} is the result of the convolution between the PSF and the perfect image of the object S_{obj}:

$$S_{deg} = S_{real} * PSF$$

According to the convolution theorem:

$$\mathscr{F}(PSF * S_{real}) = \mathscr{F}(PSF) \cdot \mathscr{F}(S_{real})$$

Then, the degraded image can be expressed as:

$$S_{deg} = \mathscr{F}^{-1}(\mathscr{F}(PSF) \cdot \mathscr{F}(S_{real}))$$

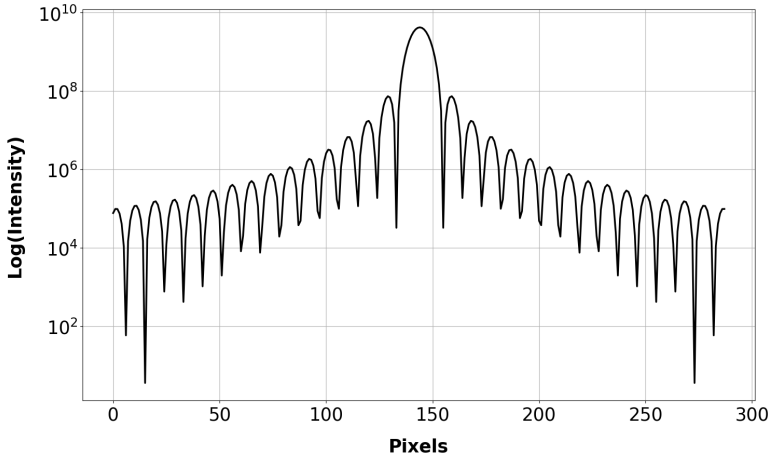


Figure 4: Point spread function in logarithmic scale of a 50 cm diameter telescope asumming a pixel size of 0.0287 arcsec.

The PSF used is shown in Figure 4. All the 61 images of the four Stokes parameters are subjected to this transformation. The result can be appreciated in Figure 5.

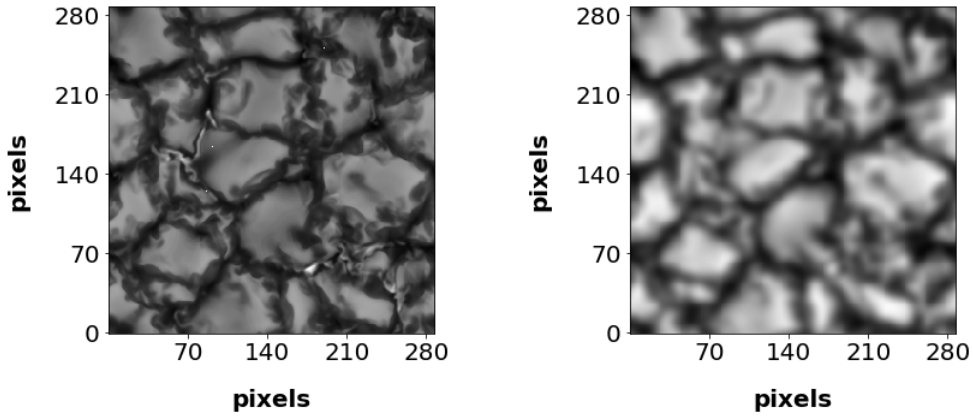


Figure 5: In the left panel, an original image of I is depicted. The right panel shows the same image after degrading it.

4.2.2 Principal component analysis (PCA)

As it has been previously recalled, the samples from the set of magnitudes and profiles consist of 427 (7 x 61) and 2404 (4 x 601) values respectively. These quantities could be the input and expected output of the model straight away, but usually, the data is correlated in some amount and not all the values are needed to grasp the main features of what they represent.

A very popular and powerful technique to drastically compress the amount of data while preserving the most relevant information is the PCA decomposition. Through an orthogonal linear transformation, the samples are projected onto a new coordinate system. The first coordinate (the first principal component) must retain the greatest variance. The next PC, similarly, must retain the largest variation of the residual subspace and be orthogonal to the previous one, and so on with subsequent PCs. Like so, the components are ordered by the amount of information they keep, making it easy to discard dimensions of low relevance and end up with just a fraction of the original values without compromising the fundamental behaviour of data. Also, having uncorrelated inputs and expected outputs helps the ANN map both sets better.

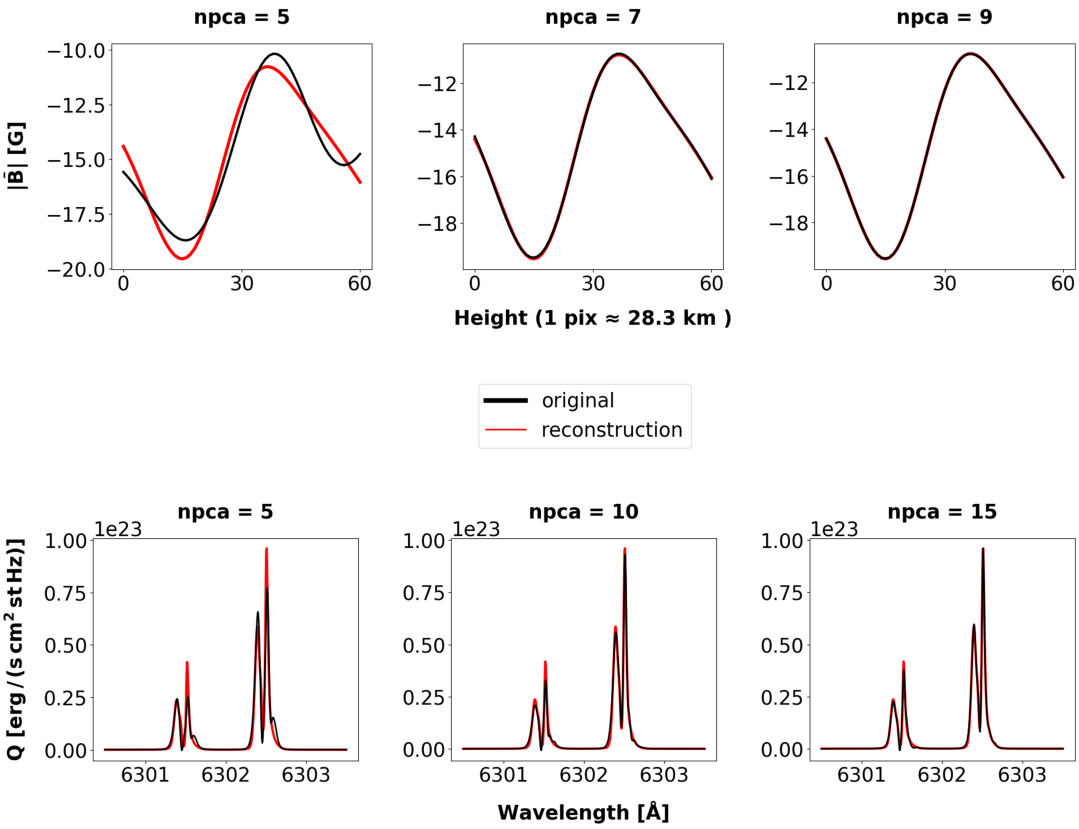


Figure 6: A sample from the magnetic field strength and from the Stokes parameter Q are chosen to find the minimum number of PCAs (npca) needed to recreate the original data adequately.

As it is displayed in Figure 6, only 8 and 12 PCs are needed to recreate fairly well the physical conditions and the Stokes profiles respectively. Now, the samples consist of 56 (8 x 7) and 48 (12 x 4) values.

There is a significant difference between fitting the original and degraded data. Note in Figure 5 how the degraded image has lost contrast, i.e. the intensity distribution is more homogenous, which

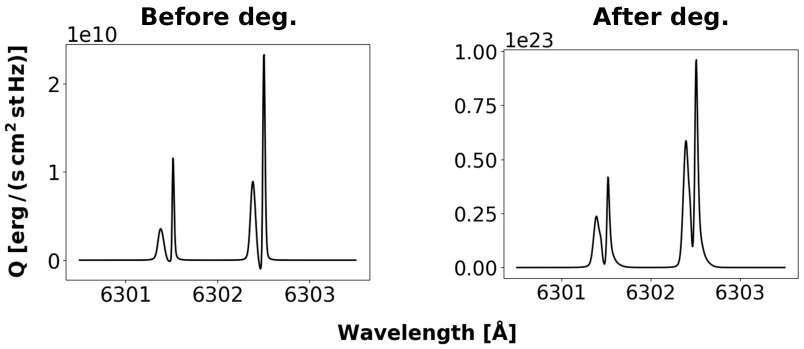


Figure 7: Sample of the Stokes parameter Q before and after degrading the data.

make the relative change between the two lines less abrupt (Figure 7). Also, subtle broadening can be noticed. This helps the new basis recreate well the original profile with fewer components.

4.2.3 Rescaling

A usual step in Machine Learning regarding data preparation is rescaling the variables before feeding them to the model. There are several possible transformations that would help the training process, for example, standardization, which yields variables with mean equal to 0 and standard deviation equal to 1 (useful for distributions close to Gaussians), or normalization, that limits the range to [0,1] without changing the distribution. A version of the latter could limit the range to [-1,1], thus centering the data.

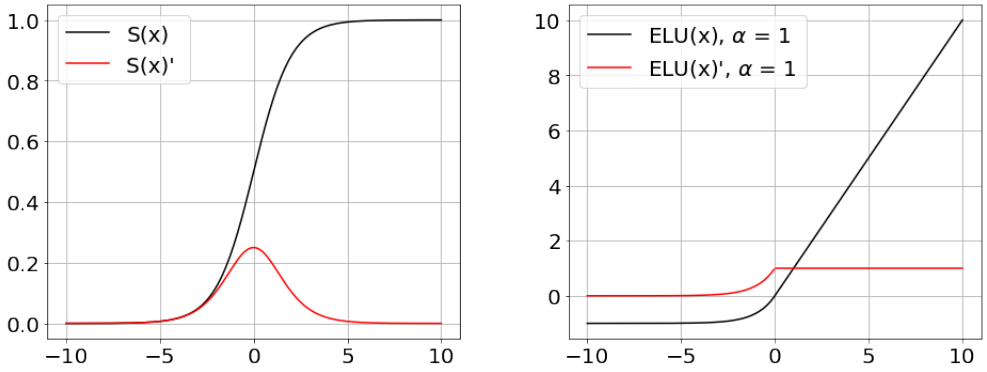
The most effective transformation depends highly on the problem; not only on data but on the whole model and optimization technique too [Shanker et al., 1996]. It has been found that in this particular case, the best results are returned when the quantities are divided each by a number so that the quotient lies approximately within the interval [-1, 1].

	T	P	V_{LOS}	V_{mic}	$\lvert\bar{B}\rvert$	γ	θ
Factor	$4 \cdot 10^4$	$8 \cdot 10^5$	$6 \cdot 10^6$	20	$2 \cdot 10^4$	16	20
Max. coeff. (approx.)	0.128	0.133	0.986	$2.914 \cdot 10^{-15}$	1.218	0.334	0.575
Min. coeff. (approx.)	-1.224	-1.046	1.197	-1.172	-1.184	-1.236	-1.220

	I	Q	U	V
Factor	$3 \cdot 10^{27}$	$1 \cdot 10^{25}$	$1 \cdot 10^{25}$	$3 \cdot 10^{25}$
Max. coeff. (approx.)	0.160	0.322	0.704	0.662
Min. coeff. (approx.)	-1.465	-0.775	-0.650	-0.788

Table 1: Factors by which each quantity is divided, and the maximum and minimum value it reaches afterward. These should be around 1 or -1.

Rescaling can be crucial under some circumstances. If some variables exhibit a higher order of magnitude than others, they will have more influence over the result due to its larger value, even if they are all equally important predictors. Also, considering that matrix multiplication is the main operation in ANNs, products between very big or very small values cause the parameters to explode to infinity or vanish to 0, making the model useless. In addition, relatively high absolute values can lead to the vanishing gradient problem where weights are no longer updated. Some activation functions present saturation areas for extreme values, like the sigmoid function shown in Figure 8, where changes in the argument yield little to no change in the result. So when a value stored in a neuron lies in these areas, incrementing differentially the weights and bias associated won't really change the result returned by the activation function, meaning that the gradients are close to zero and the parameters won't be adapted.



$$S(x) = \frac{1}{1 + \exp(-x)}$$

$$ELU(x) = \max(0, x) + \min(0, \alpha(\exp(x) - 1))$$

Figure 8: The sigmoid (left panel) and the exponential unit function (right panel) are often used as activations in Machine Learning. Their equations are indicated at the bottom and their first derivatives are represented in red.

4.3 Architecture

The ANN structure used in this project is shown in Figure 9. It consists of linear nodes arranged in 12 layers, where each node is connected to all the others located in the previous and (or) next layer. The information moves from the input to the output only in one direction, forward. It is, then, a fully connected multi-layer feedforward network.

As activation function, the "exponential linear unit" (ELU) will be considered (Figure 8). It alleviates the vanishing gradient problem as the derivative tends to zero only for negative arguments, and speeds up learning compared to other similar functions [Clevert et al., 2015]. The ELU will be applied after every layer, except the last one.

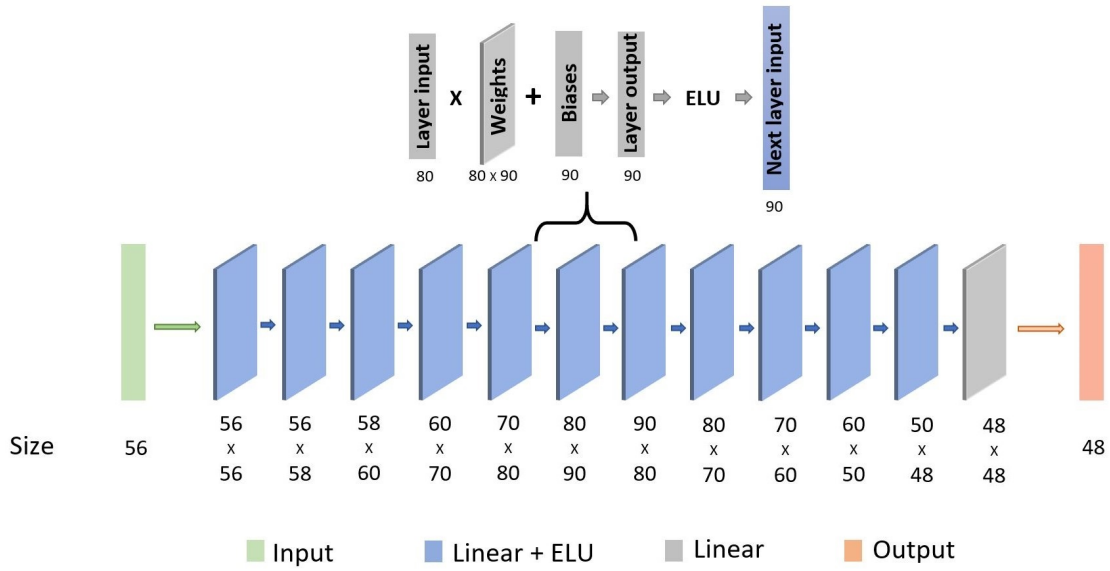


Figure 9: ANN architecture, conformed by a series of linear layers with different sizes and the application of an activation function (ELU) to every layer output, except the last one. One of the components is chosen to further detail the operations that take place between one layer input and the next.

The model parameters can be initialized randomly, following some distribution or restricting them to a range. Starting with too large or too small parameters potentially leads to the same problems as not rescaling the data (section 4.2.3), so boundaries are recommended. The Kaiming initialization [He et al., 2015] will be applied, as it is tailored for deep ANNs that use asymmetric, non-linear activation functions like ELU. Then, the weights will follow a zero-mean Gaussian distribution whose standard deviation is $\sqrt{2/n_l}$, where n_l is the number of values coming into a given layer l from the previous layer output, and the biases will equal 0.

4.4 Training process

The first 80% of the 288x288 = 82944 samples in the dataset will be used as the training set and the remaining 20% as the validation set, which will be fed to the NN in smaller batches. The former will be used to update the model parameters while the latter to evaluate how the model performs on unseen data. If the training loss is much lower than the validation loss, the model would be overfitting, meaning that it is learning to reproduce really well the samples used in the training but doesn't generalize well to new data. The other way around, it would be underfitting, not learning properly from the training data. Thus, similar losses are desired after every iteration.

The mean squared error (MSE) will act as loss function, which is adequate for regression problems where data is free of outliers:

$$MSE_{loss} = mean(L);\ L = \{l_1, l_1, \ldots, l_N\},\ l_n = (X_{NN} - X_{data})^2\,,$$

where N is the batch size.

The algorithm known as Adam [Kingma and Ba, 2014] will be used to optimize the model parameters, whose name is derived from adaptive moment estimation. In contrast to other optimizers, Adam computes individual learning rates for every parameter, and these are dynamically adapted using estimates of first and second moments of the gradients as the learning process unfolds, thereby accelerating convergence. In addition, it works well across a wide range of deep learning architectures and its hyper-parameters typically require little tuning. These hyperparameters are set to:

$$learning\ rate = 3 \cdot 10^{-4},\ \beta_1 = 0.9,\ \beta_2 = 0.999,\ \epsilon = 10^{-8}$$

The training algorithm, in each iteration, will take a batch of samples from the training set, feed it to the ANN, compare its output with the expected one via the loss function, update the weights and biases accordingly and carry out the same procedure with a batch of samples from the validation set without updating the parameters.

The ANN will be built and trained for 1189 epochs in batches of 5210 samples using Pytorch as framework [Paszke et al., 2019].

4.5 Inversion method

A random sample is taken from the validation set of Stokes profiles, representing an observation a telescope could have taken. Now, the objective is to infer the atmosphere state that produced that observation.

First, a vector full of zeros of size 56 (8 x 7) is created. In the first iteration, this vector will be fed to the ANN, whose parameters are fixed. The output will be compared with the observation via the MSE function. Then, the optimizer Adam is applied over the input vector. It will calculate the gradient of the error with respect to the values of the input vector, and update these accordingly. This is repeated until a threshold is reached.

The optimization has been carried out in the course of 158839 iterations with the following hyper-parameters for Adam:

$$learning\ rate = 2 \cdot 10^{-2},\ \beta_1 = 0.9,\ \beta_2 = 0.999,\ \epsilon = 10^{-8}$$

5 Results and discussion

En esta sección se adjuntan los resultados y discusión del entrenamiento de la ANN por un lado, y de la inversión de un perfil de Stokes por otro. Se presenta el error de validación y de entrenamiento, junto a tres ejemplos mostrando la salida de la ANN y el perfil de Stokes que trata de ajustar en cada caso. Además, se realiza un estudio estadístico básico, calculando la diferencia entre la salida de la red y el perfil real para luego obtener su media y los percentiles 5, 16, 50, 84 y 95 para cada longitud de onda. En el caso de la inversión, se muestra el error entre las magnitudes atmosféricas recuperadas y las reales, representando además ambas curvas y las de los perfiles de Stokes correspondientes.

5.1 ANN training

Analysing the loss behaviour throughout the epochs (Figure 10), it can be concluded that the training has unfolded adequately. A steep slope characterises the firsts epochs, which decreases gradually until convergence is reached. A closer look reveals small oscillations, considered normal in an optimization as long as the loss presents an overall decrease. Also, there is no evident overfitting or underfitting, as both the validation and training loss evolve similarly and not far from each other.

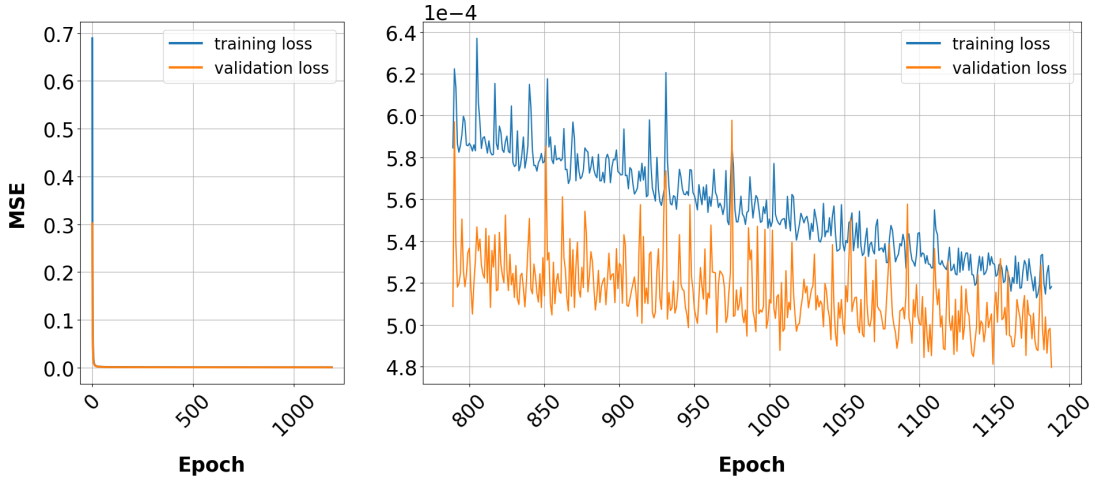


Figure 10: The training and validation loss are depicted. The left panel shows the overall behaviour of the error, while the right panel focus on the last 400 epochs, displaying a more detailed representation of it.

After some tests, it was noted that the validation loss would reach its lower value at $4.8 \cdot 10^{-4}$ and oscillate around a higher one in subsequent epochs, so the training was stopped when that minimum had been reached. Note that the MSE is calculated over expected outputs that mostly lie in the range [-1, 1].

Not all samples are equally adjusted by the ANN, as can be extracted from Figures 11a, 11b and 11c. The performance varies considerably from one sample to the other (e.g., the ANN outputs in Figure 11a are, in general, closer to the expected profiles than those in Figure 11c), but also from one Stokes parameter to the other (e.g., in Figure 11a, the expected I, Q and U are reproduced quite well while V differs substantially). However, all three samples agree with the fact that the best results are obtained for I. The parameters Q, U and V present very different shapes and peak values in the dataset used, which the ANN tries to account for, making the task of mapping the two sets more difficult.

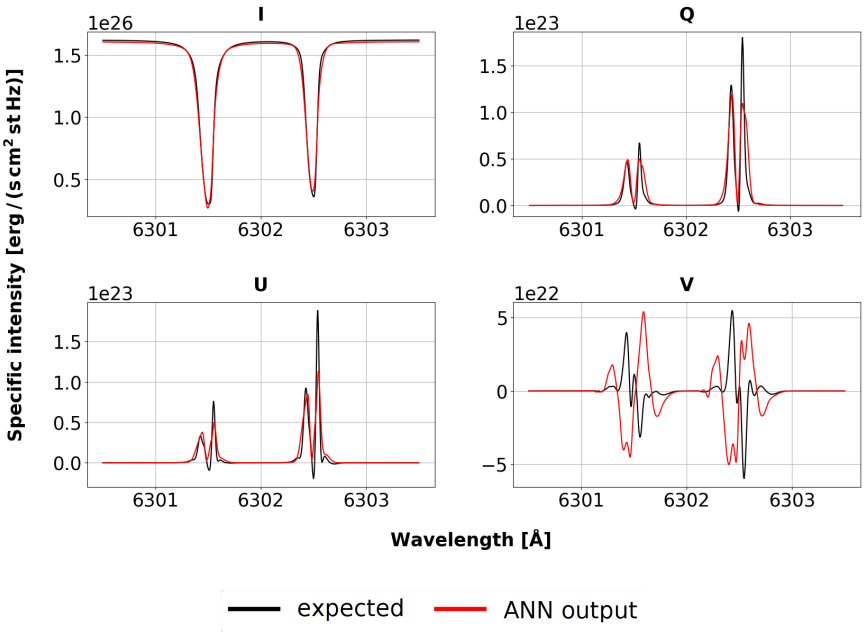


Figure 11a: Comparison between an expected Stokes profile sample and the one given by the ANN.

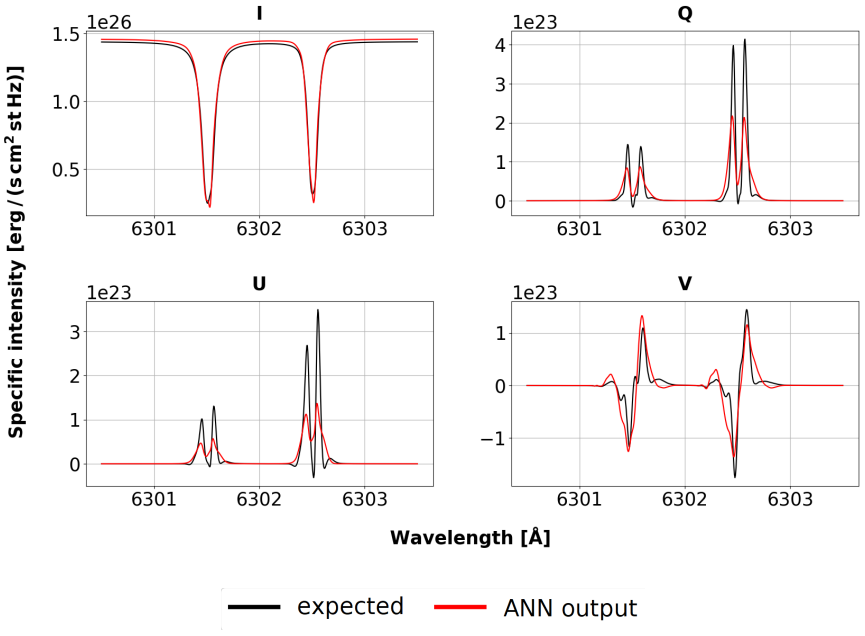


Figure 11b: Comparison between an expected Stokes profile sample and the one given by the ANN.

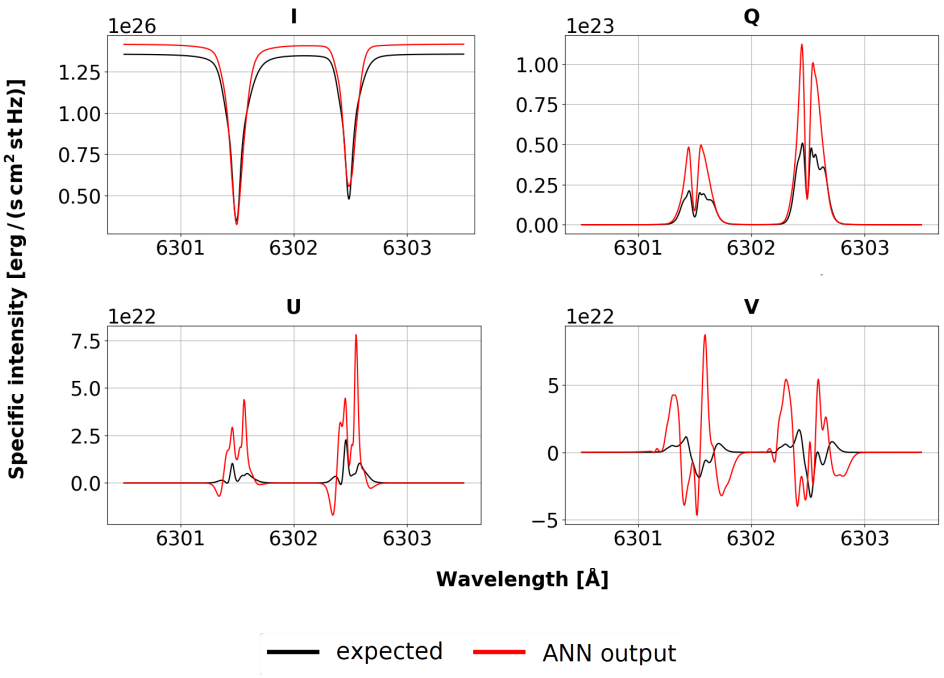


Figure 11c: Comparison between an expected Stokes profile sample and the one given by the ANN.

To better understand how serious the discrepancies are, a basic statistical study is carried out over the absolute difference between the expected profiles and the ANN output. Figure 12a and 12b reveal mean variations that reach the same order of magnitude than the quantity they are attributed to. The parameter I shows a lower mean at its peak values, 6301.5 Å and 6302.5 Å, while it is much higher around them. This phenomenon can also be visualized in Figure 11: I peaks are generally well adjusted while the rest of the curve differs more. It happens the other way around with Q and U, as its peaks are narrower than those in I, making it harder for the ANN to reproduce them. The mean variation in the parameter V is more homogeneous as its shape varies greatly from sample to sample. The percentiles follow the same behaviour described for the mean. Considering the scale of the graphs and that the order of magnitude of I is around $10^{25} - 10^{26}$ and that of Q, U and V around $10^{22} - 10^{23}$, it can be concluded that only in 5% of the validation samples, the differences are between one and two orders of magnitude less than the parameters. The distribution of differences in V is particularly asymmetrical, as almost 84% of the values can be encompassed under the mean, in a narrow range, while a larger range is needed to include the remaining 16%.

As has been discussed, the samples of Q, U and V present very different shapes. Based on the better results obtained for I, it is reasonable to assume that having more data, i.e. more samples that resemble each shape, would help the ANN learn to reproduce each profile more adequately.

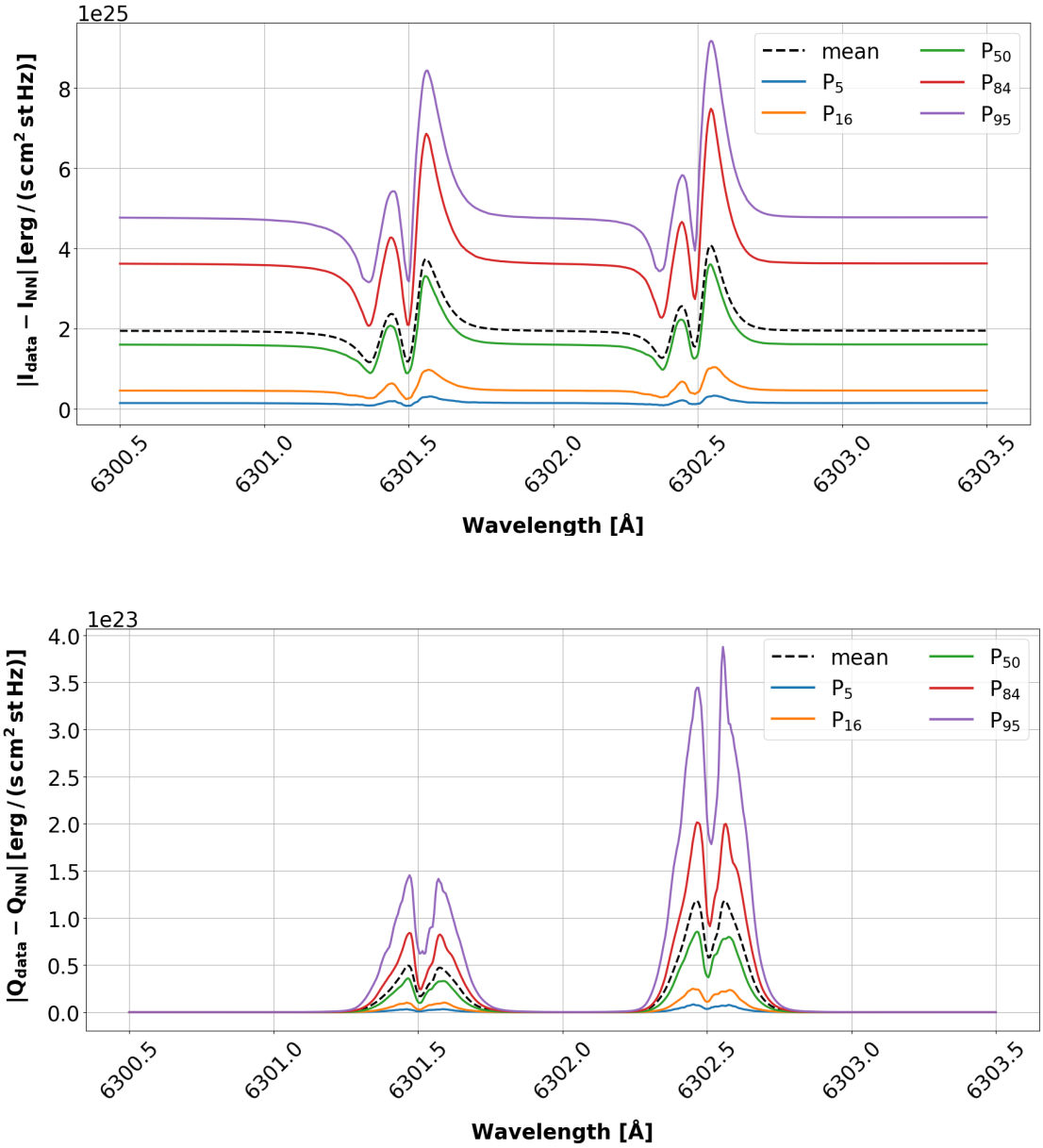


Figure 12a: Mean and percentiles 5, 16, 50, 84 and 95 of the absolute difference between the expected profiles from the validation set and the ANN output. These quantities are calculated for every wavelength.

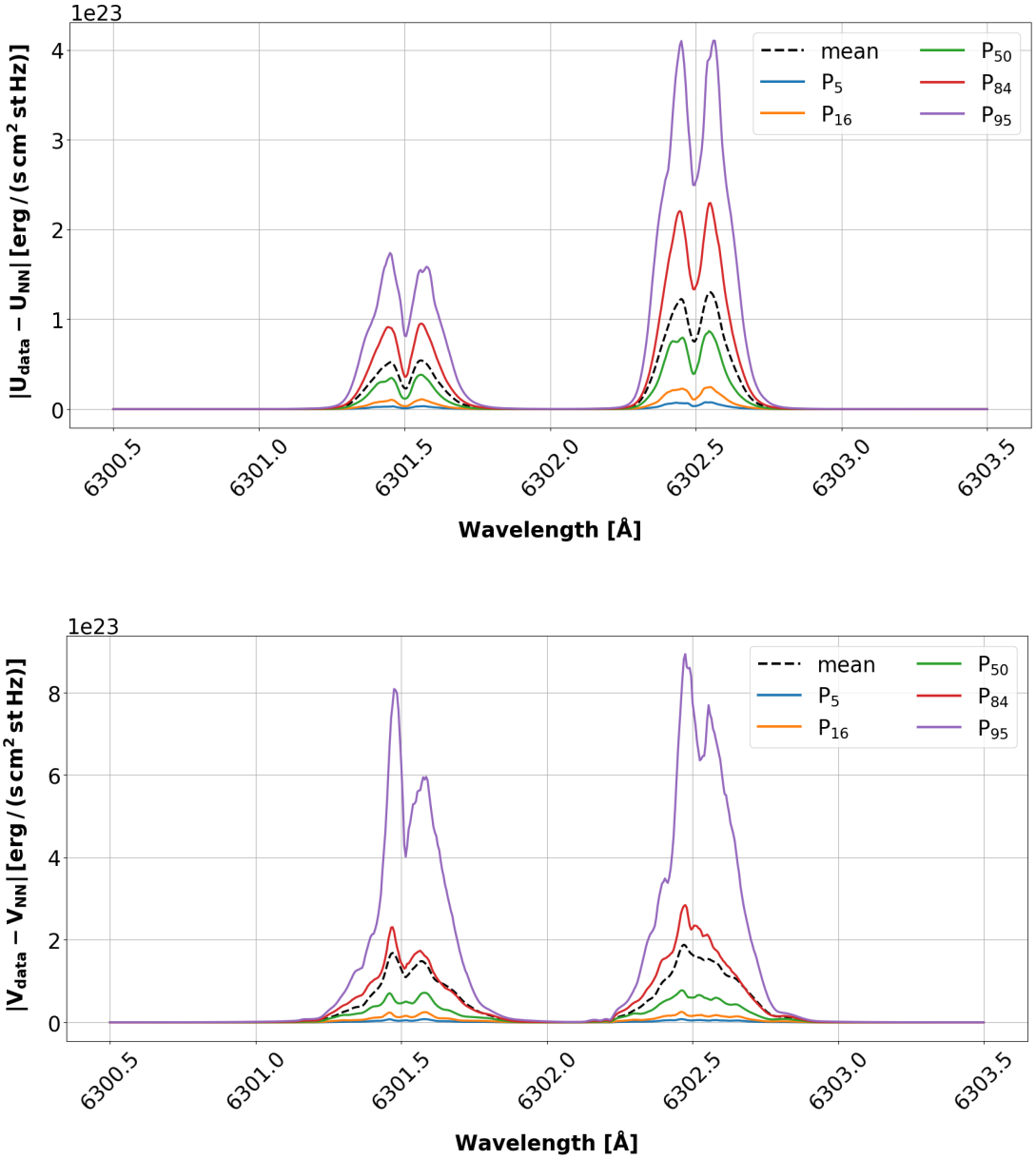


Figure 12b: Mean and percentiles 5, 16, 50, 84 and 95 of the absolute difference between the expected profiles from the validation set and the ANN output. These quantities are calculated for every wavelength.

5.2 Inversion

Similarly to the ANN training, the optimization stopped when the loss started converging and reached a minimum value. The oscillations shown in Figure 13 resemble those seen in Figure 10 (note that the scale is one order of magnitude greater in that graph), and the loss descends abruptly in the firsts iterations as it should. In the last 1000 iterations, there is no evident change overall, i.e. it has reached convergence. Still, it can be seen in Figure 14 that the profiles fitting is insufficient, as can be expected after the discussion in section 5.1.

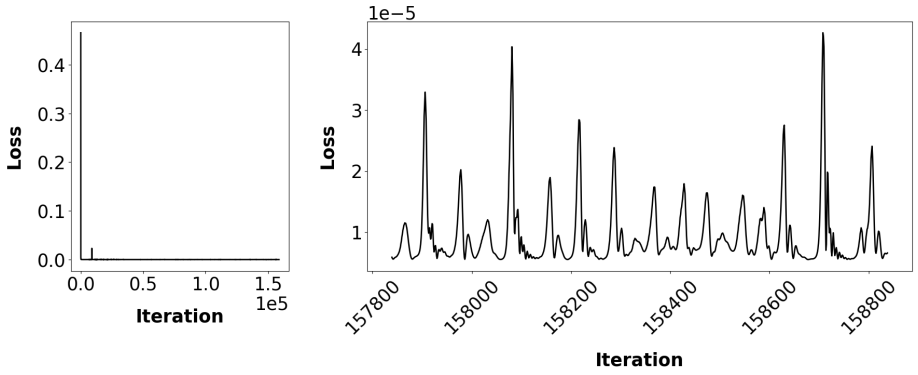


Figure 13: The loss calculated in the inversion process is depicted. The left panel shows the overall behaviour of the error, while the right panel focus on the last 1000 iterations.

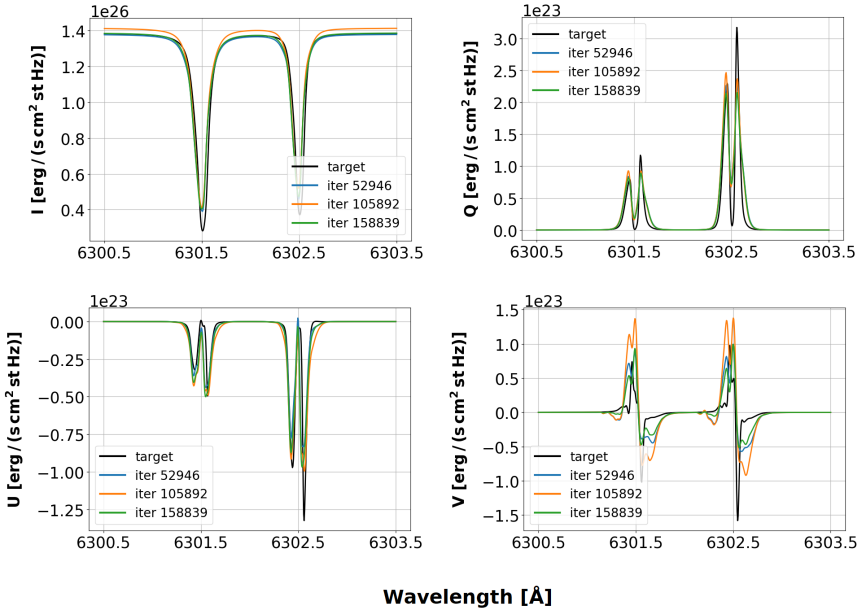


Figure 14: Comparison between the observation and the ANN output at three different iterations.

The magnitudes obtained after the optimization, shown in Figure 15a and 15b, don't resemble the expected ones at all. It is not possible to infer magnitudes similar to the originial ones when the corresponding profile differs substantially from the one given by the ANN. The problem is too degenerated.

These results could experience a slight improvement using another optimization method, one more suitable for this problem. A viable alternative would be to implement the Levenberg–Marquardt algorithm [Levenberg, 1944, Marquardt, 1963], used to solve non-linear least squares problems.

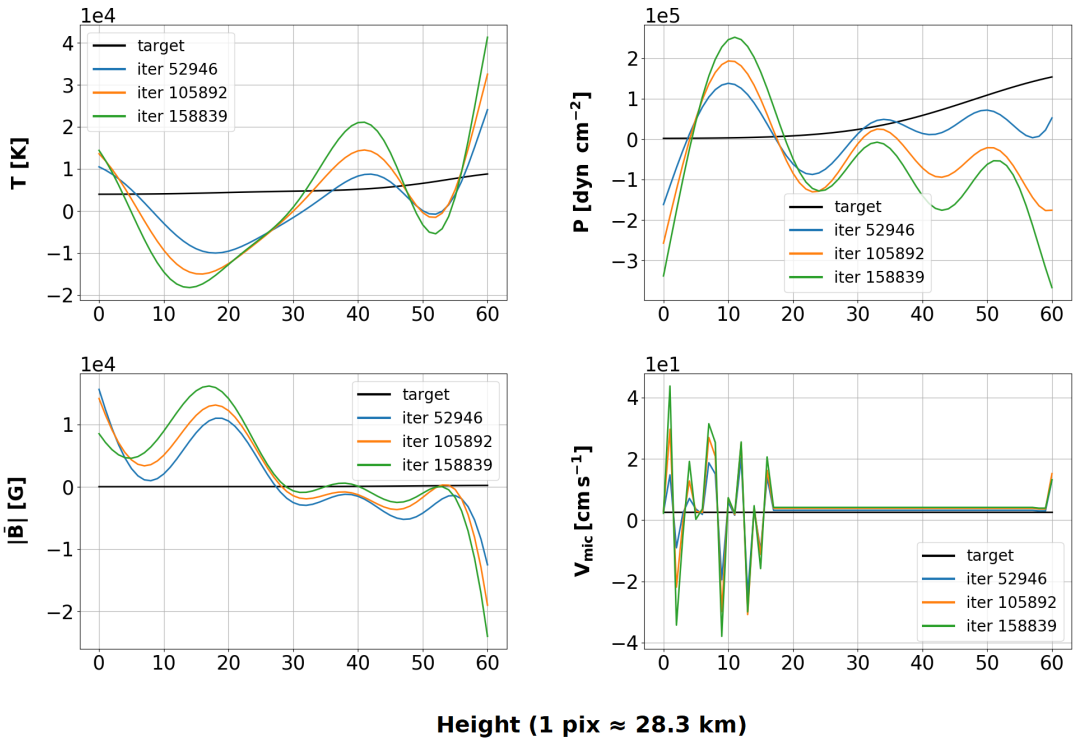


Figure 15a: Comparison between the expected T, P, $|\bar{B}|$ and V_{mic}, and the input vector at three different iterations.

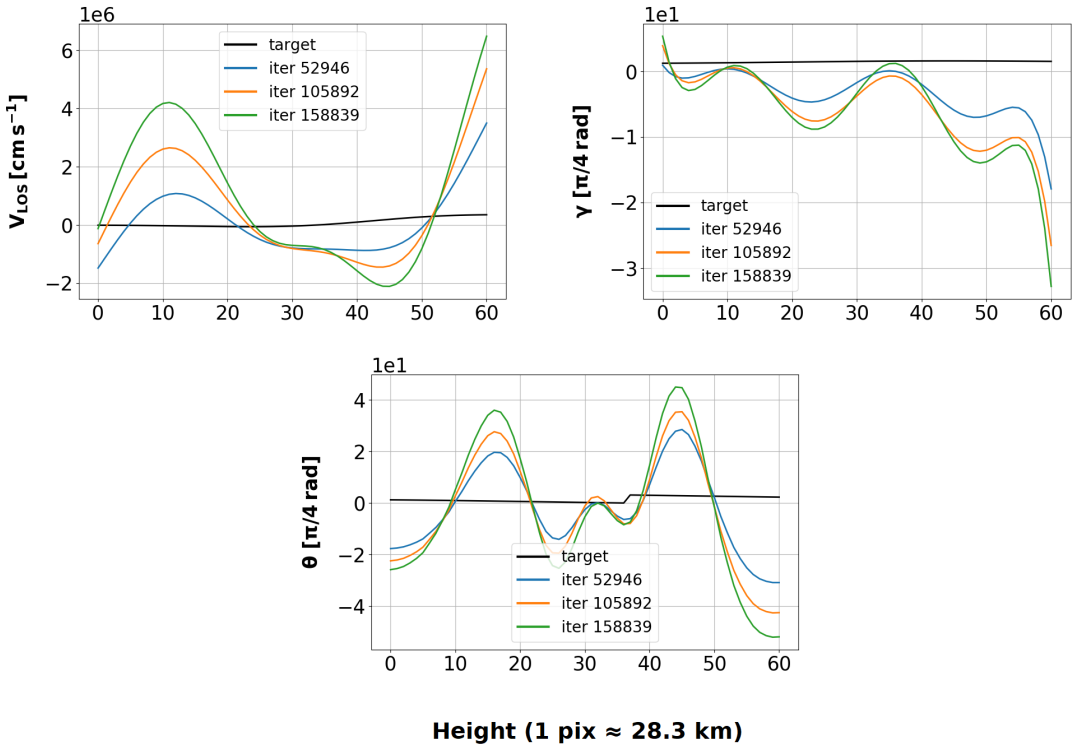


Figure 15b: Comparison between the expected V_{LOS}, γ and θ, and the input vector at three different iterations.

6 Conclusions

Finalmente, se ofrece una visión global del trabajo llevado a cabo, junto a un resumen de los resultados más relevantes y de las conclusiones extraídas a partir de los mismos.

In this project, we have explored the application of an artificial neural network on a Solar Physics problem, the inversion of Stokes profiles.

An introduction to ANNs and their training process has been made, followed by a description of the inversion problem, where the motivation for trying a different approach was recognised. The provided data was subjected to degradation, principal component analysis and rescaling before feeding it to the ANN. The configuration of the ANN and the optimization method were chosen in accordance with the characteristics of the problem at hand.

Although the training went satisfactorily and some good results were retrieved, the model could not adjust decently most of the profiles, with differences of the same order of magnitude than the parameters. As it was acknowledged that the parameters with greater variation in shape from sample to sample were the worst fitted, the need for more data was suggested to improve the ANN

performance.

Finally, the inversion of one of the profiles samples was attempted using the same optimization method as in the ANN training. The inferred magnitudes disagreed completely with the expected ones, as the problem was too degenerated. The Levenberg–Marquardt algorithm was suggested as an alternative optimization method that could yield slightly better results.

7 Bibliography

S. Chandrasekhar. *Radiative Transfer*. Dover Publications, 1960. ISBN 0-486-60590-6.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015. [arXiv].

José Carlos del Toro Iniesta. *Introduction to Spectropolarimetry*. Cambridge University Press, April 2003. ISBN 0521818273. [ADS].

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, 2015. The IEEE International Conference on Computer Vision (ICCV). [arXiv].

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015. [arXiv].

Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, 2(2):164–168, 1944. [AMS].

Giuseppe Longo, Erzsébet Merényi, and Peter Tino. Foreword to the focus issue on machine intelligence in astronomy and astrophysics. *Publications of the Astronomical Society of the Pacific*, 131:100101, 11 2019. [arXiv].

Donald W Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, 11(2):431–441, 1963. [SIAM].

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. [URL].

Murali Shanker, Michael Y Hu, and Ming S Hung. Effect of data standardization on neural network training. *Omega*, 24(4):385–397, 1996. [ScienceDirect].

H. Socas-Navarro, J. de la Cruz Rodríguez, A. Asensio Ramos, J. Trujillo Bueno, and B. Ruiz Cobo. An open-source, massively parallel code for non-lte synthesis and inversion of spectral lines and zeeman-induced stokes profiles. *Astronomy & Astrophysics*, 577:A7, Apr 2015. ISSN 1432-0746. doi: 10.1051/0004-6361/201424860. [A&A].

G. G. Stokes. On the composition and resolution of streams of polarized light from different sources. *Transactions of the Cambridge Philosophical Society*, (9):399, 1852.

Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. 2013. New York University. [arXiv].