Highlights of Spanish Astrophysics XI, Proceedings of the XV Scientific Meeting of the Spanish Astronomical Society held on September 4–9, 2022, in La Laguna, Spain. M. Manteiga, L. Bellot, P. Benavidez, A. de Lorenzo-Cáceres, M. A. Fuente, M. J. Martínez, M. Vázquez- Acosta, C. Dafonte (eds.), 2023

Do machines dream of modelling AGB stars?.

Santander-García, M.[1], **Manuel Julián, J. A.**[1,2], **Alcolea, J.**[1], **Bujarrabal, V.**[1], **and Asensio Ramos, A.**[3,4]

[1] Observatorio Astronómico Nacional (OAN-IGN), Alfonso XII, 3, 28014, Madrid, Spain
[2] Universidad Internacional de Valencia (VIU), c/del pintor Sorolla, 21, 46002, Valencia
[3] Instituto de Astrofísica de Canarias, c/Vía Láctea s/n, 38205 La Laguna, Tenerife
[4] Departamento de Astrofísica, Universidad de La Laguna, 38206 La Laguna, Tenerife

Abstract

It is very common in astrophysics that certain relevant parameters of the objects studied cannot be obtained directly from observations, but require numerical models that simulate the relevant physical mechanisms, often through an iterative process of trial and error. This is often lengthy and consumes a large part of the human and material resources of the research process. This is the case in the characterisation of the circumstellar envelopes of Asymptotic Giant Branch (AGB) stars, which could greatly benefit from the use of Artificial Intelligence Deep Learning (DL) techniques. We present the preliminary results of this project as an illustrative example of what can be achieved by (and expected from) the application of DL to numerical modelling in astrophysics.

For this project we have trained a convolutional neural network (CNN) to predict the physical conditions of these envelopes (mass loss, molecular abundance, temperature distribution, expansion velocity pattern, and distance) from a reduced set of single-dish observations of ^{12}CO rotational transitions including J=1–0, J=2–1, as observed with IRAM 30m, and J=6–5, J=10–9, and J=16–15 as observed with HERSCHEL/HIFI. The network has been trained by building a complete library of numerical models covering a large range of parameters using a custom radiative transfer 1D code, which computes the excitation of CO using the Large Velocity Gradient (LVG) approximation and then solves the radiative transfer problem via ray tracing, thus generating synthetic CO profiles from a detailed description of the physical structure of a circumstellar envelope. Once the learning is completed, the system should be able to determine these fundamental parameters from observations without the need for 'manual' model fitting.

1 Introduction

The goal of this work is to evaluate the feasibility of using Deep Learning (DL) techniques in the study and characterization of evolved star envelopes, specifically stars in the Asymptotic Giant Branch (AGB) phase. The objective is to determine if we can predict the most important physical parameters of these systems from observations in radio of certain emission lines of the CO molecule. The specific goal is to characterize the envelope, determining its mass loss rate, the terminal velocity of the gas, the spatial distribution of the velocity and temperature, the CO abundance, and the distance to the star.

The preliminary work presented here will be further discussed in the Master's Thesis by one of the authors, José Antonio Manuel Julián, to be presented in early 2023.

2 Artificial neural networks

In the last 10 years significant progress has been made in the development of Artificial Neural Networks (NN), due to advancements in computing capacity and the growing trend in the availability of public data sets. A NN is defined as a set of nodes, called neurons, connected to each other that apply a non-linear transformation to input data to produce an output data. The original idea of NNs dates back to the late 1950s with the invention of the perceptron by Frank Rosenblatt [5], but could not be properly developed until decades later due to the limitations of early computers.

Just like in a biological neuron, an artificial neuron receives multiple inputs, from all of the outputs of the neurons it is connected to, it applies weights to those inputs, also applies an activation function, usually non-linear, and generates a specific output that is propagated to the neurons connected to its output. Normally the different neurons are organized in layers, where neurons only connect from one layer to another layer. This way, we will have an input layer, an output layer, and in between a series of layers that we usually call hidden layers. Depending on the number of hidden layers, we will have deeper or shallower networks. Signals are propagated from the input layer to the final output layer where the desired results are obtained.

One of the most successful applications of NNs has been image processing. This has been possible thanks to the development of a neural network architecture called Convolutional Neural Network (CNN), which are able to extract specific features from images and establish correlations between them, as well as their spatial distribution in the image by applying convolutional filters. However, the price to be paid is that they are much deeper and require more computing and memory capacity. In recent years, various companies dedicated to the development of artificial intelligence, such as Google, Facebook, etc., have developed different CNN architectures, training them with image datasets and making them available to the community. Some of them have represented a notable advance in the analysis and identification of objects in images and video.

Basically, the operation of a CNN is to accept a data tensor as input (the RGB values at each pixel of the image), apply a series of convolutional filters to that input, extract a feature map of the image and establish relationships between them, using fully connected neuron layers as output. Some of the most popular CNNs are capable of recognizing objects in images as well or even better than a human, for certain very specific tasks for which they have been conveniently trained.

The training process of an NN is very demanding in terms of computation and memory, as it consists in minimizing an error function between the output of the network and the expected output. The error is calculated through an equation and it is sought to minimize it by adjusting the weights of the connections between the neurons using the gradient descent method. This process is repeated until the errors are small. The use of graphic cards (GPU) has significantly improved the training time, as they are specially designed to handle tensor operations and can improve performance by a factor of 20.

3 Methodology

The main idea behind this work is to take advantage of the success of CNN networks, specifically those that have already been shown to be efficient in extracting and identifying patterns in images. By using a pre-trained CNN network, it is expected that the training performance will be improved so that it can learn new patterns that will identify our images.

The goal is to perform simultaneous fitting of sets of single-dish observations of ^{12}CO rotational transitions including J=1–0, J=2–1, as observed with IRAM 30m, and J=6–5, J=10–9, and J=16–15 as observed with HERSCHEL/HIFI. The data of each set of lines is transformed into an image where the most outstanding line features can be seen thus taking advantage of the power of CNN networks. A large set of images, built from synthetic modelling by a 1D radiative transfer code using the Large Velocity Gradient (LVG) approximation [1], are used to train the network by associating each image with a point in the physical parameter space. The challenge now is to create an appropriate image from the line profiles. To do this, a 3D surface is created by representing the lines in that 3D space, repeating each line 10 times, and placing each set of repeated lines one after the other. A gap of 3 additional values, whose values are the interpolation of the data, is added between each block of repeated lines in order to achieve a smooth surface. Once the surface is created, it is projected onto the plane Z = 0 as a density map of the height of this surface. See Fig. 1 for an example of 3D surface construction.

These meta-images are associated with the set of physical parameters and are used to train the CNN network. In order to have a broader set of images, and to face the network with the noise that we will probably have in the real data obtained from a radio telescope, we generate 10 meta-images for each point in the parameter space. Each of these 10 images is the same version but adding noise to the data of each line, spanning a range of S/N similar to those typically found in real observations. This allows the CNN to learn the fundamental features that distinguish one image from another without being biased or affected by noise in the spectra.

3.1 The parameter hyperspace

The first important decision is how to discretize the space of physical parameters and how many simulations we needed to do to populate the parameter hyperspace. Initially, we considered a range of values that resulted in 10^7 simulations, where in each simulation the profile of 5 lines for 12CO and other 5 lines for 13CO must be calculated. In this scenario, despite having 16 double thread cores available from a machine at the Observatorio Astronómico

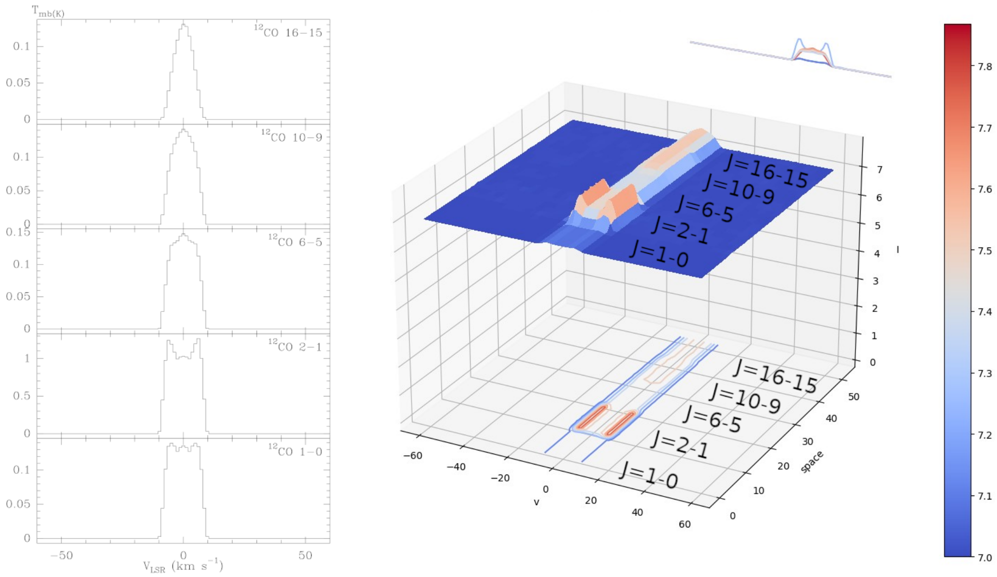

Figure 1: **Left**: Synthetic line set corresponding to a model AGB with $\dot{M}$=3.16×10^{-7} $M_\odot$ a^{-1}, $X_{^{12}CO}$=2×10^{-4},v_∞=8 km s^{-1}, α_T=0.8, β_v=2.0, D=316 pc. α_T and β_v refer to the exponent of the radial distribution of the gas temperature and velocity, respectively. **Right:** 3D surface or meta-image built by stacking each of the lines on the left, each of them cloned 10 times, with a 3-pixel interpolation between successive lines.

Nacional (OAN), and being able to run 32 processes in parallel, the time it would take for our code to perform all the scenarios was more than 30 days, just to calculate the data of the lines. To this we would have to add the time necessary to create the meta-image, which would make the total process take more than 120 days of CPU, something unviable for a Master's Thesis work.

We therefore decided to further restrict the scope of the problem and focus only on 12CO, thus reducing the number of necessary simulations and the parameter space. We priorised discretization in those parameters with a greater impact on the shape of the spectra, that is, $\dot{M}$, v_∞, $X_{^{12}CO}$, and D, ending up with a discretely randomized parameter space with fine granularity in the most relevant dimensions (see Table 1). A dataset of 290,000 images was generated to train the CNN, testing different models and modifying the final layers to adapt them to the desired output. We also introduced a modification so that the CNN would not only accept the meta-image as input, but also the maximum value of the signal present in the line data.

With regard to the CNN used, several tests were performed with different architectures, modifying the final layers to adapt them to our output. This is a common practice in DL, using a network that has already been created and previously trained, which we modify the final layers to make the output what we are looking for and retraining it with our image dataset. In this way, the network learns to extract the features of our images, in our case the meta-images.

Table 1: Range of model parameters covered. α_T and β_v refer to the exponents of the radial distribution of the gas temperature and velocity, respectively. The term 'Scale' refers to the type of distribution between the minimum and maximum values.

Parameter	Minimum value	Maximum value	# values	Scale
$\dot{M}$	10^{-7} $M_\odot$ a^{-1}	10^{-5} $M_\odot$ a^{-1}	10	Logarithmic
$\mathrm{X}_{^{12}\mathrm{CO}}$	2×10^{-4}	8×10^{-4}	10	Lineal
v_∞	8	18	10	Lineal
α_T	0.8	1.25	5	Lineal
β_v	0.8	2.0	5	Lineal
D	100 pc	1000 pc	5	Logarithmic

4 Preliminary results and conclusions

The DenseNet169 network [4] was ultimately selected for its superior performance. When training the network to focus on and predict the most relevant parameters only (i.e. $\dot{M}$, v_∞, $\mathrm{X}_{^{12}\mathrm{CO}}$, and D), the process reached an accuracy of 99.85% in sucessfully predicting the parameters of a validation set of 10,000 meta-images made from synthetic spectra lying within the sampled hyperspace volume, but never seen by the CNN before. We consider a prediction sucessful if the normalized distance between the predicted and real parameter vectors is within 10% of the normalized modulus of the real parameter vector. This training took 2 hours using GPU resources provided by the Instituto de Astrofísica de Canarias (IAC).

Figure 2 shows the relative error distribution of the validation image set in each parameter. Negative values in the figure are simply due to the way the error is defined, thus the sign can largely be ignored. As can be seen, the CNN accurately predicts v_∞ within a 6% uncertainty, while the error in the distance ranges between 10% for the smaller distances (∼100 pc) to 60% for the largest ones (∼1 kpc). The error in the ^{12}CO abundance, on the other hand, is ∼50% for abundances typical of O-rich stars, and ∼20% for those typical of C-rich stars. Finally, the CNN accurately predicts high mass losses ($\sim10^{-5}$ $M_\odot$ a^{-1}) within a ∼20%, and low ones ($\sim10^{-7}$ $M_\odot$ a^{-1}) with an expected, larger error around 100%.

Put together, these results have nothing to envy in comparison to the modelling work of an expert human (e.g. [6, 3]), which would also have taken a significantly larger amount of time. Future work, which we will detail in a forthcoming paper, consists of testing the trained CNN with data from a sample of real stars in order to evaluate its prospects in this field, including a detailed treatment of the errors and correlations between parameters following strategies tested in other fields [2]. In the meantime, in the light of present results, we can preliminary conclude that the prospects for the use of CNN for modelling AGB star envelopes are promising.

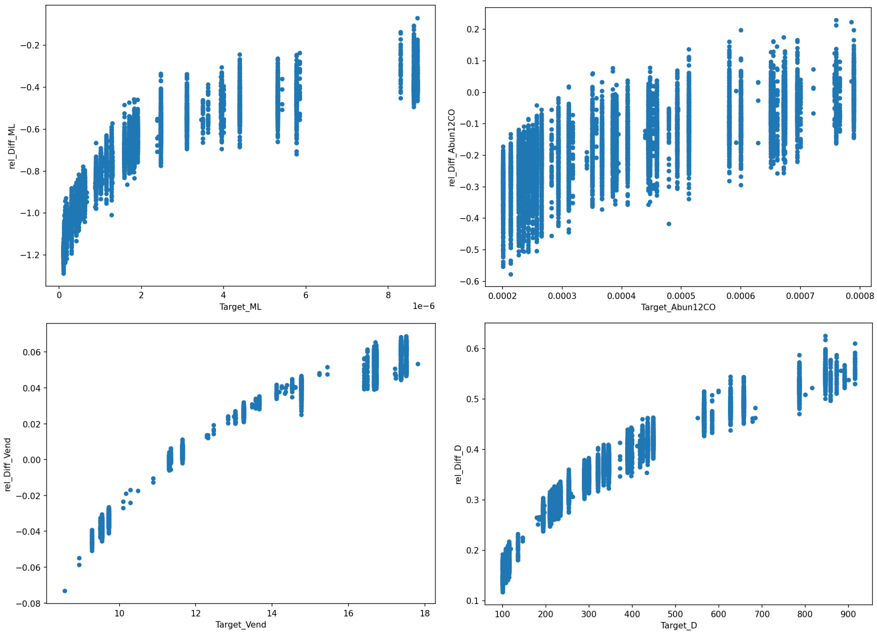

Figure 2: Relative error distribution of the image validation set in each of the most relevant parameters. From top to bottom and left to right, parameters are $\dot{M}$, $X_{^{12}CO}$, v_∞, and D.

Acknowledgments

M.S.G., J.A., and V.B. acknowledge support by the Spanish Ministry of Science and Innovation (MICINN) through project EVENTs/Nebulae-Web (grant PID2019-105203GBC21). The authors wish also to acknowledge the IAC for their kind disposition in the use of their IT resources.

References

[1] Castor, J. I., 1970, MNRAS, 149, 111

[2] Díaz Baso, C. J., Asensio Ramos, A., de la Cruz Rodríguez, J., 2022, A&A, 659, 165

[3] Díaz-Luis, J. J., et al., 2019, A&A, 629, 94

[4] Huang, G., et al., 2018, arXiv:1608.06993v5

[5] Rosenblatt, F., 1958, Psychological Review, 65, 386

[6] Santander-García, M., et al., 2012, A&A, 545, 114