Design of a machine learning model for the characterisation of young planets from dust morphologies in discs

Alessandro Ruzza

September 12, 2021

Abstract

Protoplanetary discs present substructures, such as axisymmetric regions of luminosity depletion (gaps), that can be explained by the presence of forming planets. Features of these objects can be inferred from their observation and analysis. A remarkable example is the estimation of the planetary mass from the gaps morphology. The approaches currently used, empirical formulae or numerical simulations, are both limited in precision or applicability. In this thesis we propose a machine learning approach: using a neural network to infer this information from disk images with the requirement of the least amount of physical features not directly observable. Possible future developments of such models require data for the train and test phases. We design and build a database for this purpose collecting data obtained from numerical simulations and providing an easy-to-use interface for the implementation of machine learning models using TensorFlow libraries.

Contents

Chapter 1

Introduction

Chapter 2

Protoplanetary discs

Protoplanetary discs emerge in the context of star formation. The process takes place in specific "star forming" regions filled with a mixture of gases, mainly hydrogen and helium enriched with some heavier elements, called interstellar medium. The gravitational collapse of a denser zone gives birth to a star. Gradually the star accrete its mass drawing, from its surroundings, gas and dust that organizes in an accretion disc whose shape is a consequence of angular momentum conservation. At some point, after $\sim 10^5$ yr, most of the matter has accreted into the star, the disc temperature and thermal emission are maintained by the star irradiation while gravitational collapse becomes negligible as a source of energy (passive disc). It is at this stage that the accretion disc gets identified as a protoplanetary disc due to its role as a planet formation cradle. Their study allows the development and test of theoretical models regarding planets and their genesis.

In this chapter I am going to give an overview of the structure and observable features of protoplanetary discs and of their relation with young embedded planets. The discussion will not be exhaustive and neither detailed, the aim is to give the reader a picture of these objects, and provide a basic understanding of the properties which I will be referring to in the next chapters.

2.1 Structural properties

Most of protoplanetary discs are observed at distances of about 100-200 pc in the star-forming regions. They exhibit typically a diameter of about 100 a.u. meaning that they span approximately 1 arcsec of the sky as seen from Earth. In the following sections I will use cylindrical coordinates defining the frame of reference in figure 2.1a to discuss disc properties. Only axisymmetric discs will be considered.

After their shape and location, the first question we have to address regards what discs are made of. Two main structural constituents can be distinguished according to their physical state: gas and solids. The solid component consists in dust and debris of various dimensions, going from the micrometers to a few meters, and account for about the 1% of the total mass. Despite being a small fraction of the disc, solid grains are actually easier to observe and measure through their thermal emission. Measures of their mass are, however, tied to some unknown optical properties whose estimation introduces a source of uncertainty.

Dust and solid fragments are embedded in the gaseous medium which provides the vast majority of the disc mass. Molecular hydrogen (H_2) is the main constituent which is challenging to observe due to its lack of a dipole moment. Measures related to less abundant molecules, such as HD or CO, are used to probe the properties of the gas component.

The overall mass of a typical protoplanetary disc has been measured to be some Jupyter masses. Estimates of these quantities are relevant, for example, to provide upper limits to the masses of

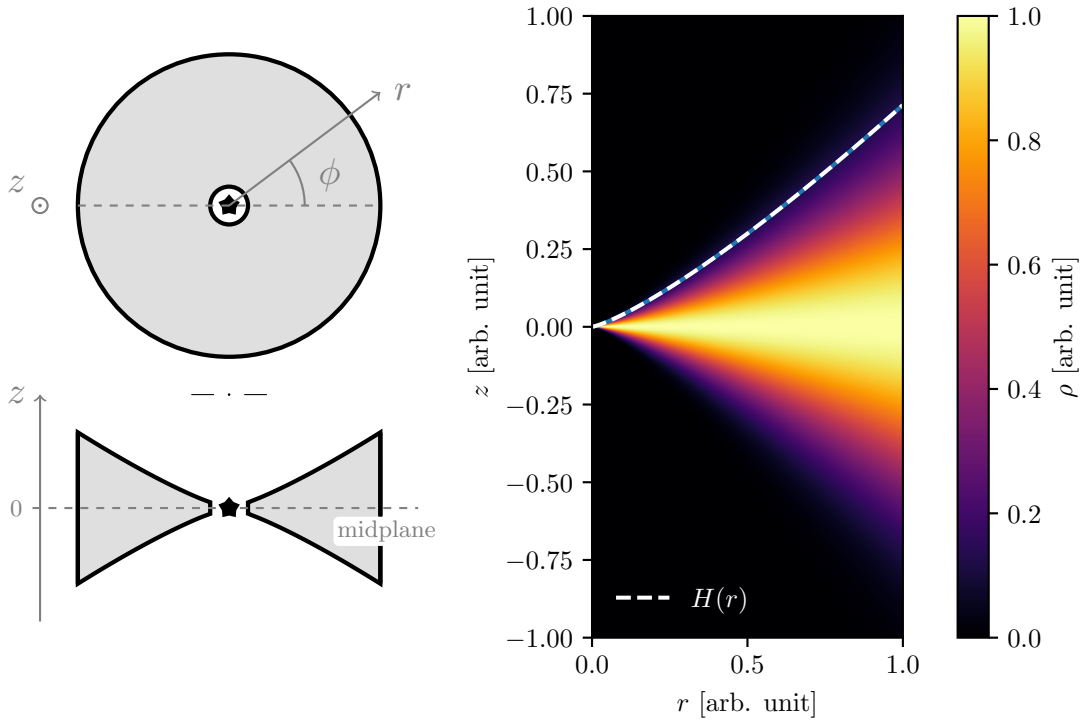

Figure 2.1: (a) left panel, scheme of the frame of reference used to refer to spatial dependencies of disc properties. (b) right panel, map of the gas density in the $r - z$ plane for a flared disk ($H(r) \propto r^{5/4}$) at vertical hydrostatic equilibrium.

forming planets.

Another important property is the gas and dust temperature, closely related to sundry factors both in the disc dynamics and radiative emission. Its value changes with the radial and vertical distance from the star going from hundreds of Kelvin to approximately 20K. The interstellar medium instead has observable features compatible with a temperature of about 10K. Protoplanetary discs are embedded in this medium that hence constitutes the background of observations and makes it difficult to reveal colder parts.

2.2 Disc dynamics

The gas and dust components just discussed, evolve and settle in characteristic structures, which I am now going to discuss.

The disc evolution can be properly described by the laws of fluidodynamics in the gravitational potential of the central star. Some assumptions need to be made in order to acquire a predictive model of practical use. The first one is called "thin-disc approximation" which consists in assuming that the radial distance is greater than the vertical typical length scale H, thus requiring $H/R \ll 1$. This quantity, called aspect-ratio, has been measured, showing values in the range $10^{-3} - 10^{-1}$ which justify the assumption. The thin disk approximation allows the study of disc properties integrating the equations along the vertical direction. The second simplification that is usually made consists in neglecting self-gravity. The stability condition of the disc against self-gravity can be written in the form

$$\frac{M_{disk}}{M_{star}} \lesssim \frac{H}{R} \tag{2.1}$$

which is well satisfied in the late epochs when protoplanetary discs are studied.

Keeping in mind these assumptions, I am going to further explore how fluid dynamics can be applied to model the disc structure and its internal motions. First, we have to focus on the gas component which is modelled as an ideal fluid characterized by the equation of state

$$P = \frac{k_B T}{m_p \mu} \cdot \rho \equiv c_s^2 \cdot \rho \tag{2.2}$$

where μ represents the mass of a single gas molecule expressed in masses of the proton (m_p).

Investigating the gas component is sufficient to predict most of the macroscopical features of the discs structure due to its predominance over the solid elements.

The vertical structure of the gas is determined by a steady-state solution of the hydrodynamical laws of motion together with the Poisson equation that accounts for the gravitational potential. The thin-disk assumption allows a great simplification of this problem leading to the vertical density profile

$$\rho(z, r) = \rho_0(r) \exp\left(-\frac{\Omega_K^2 z^2}{2c_s^2}\right) \equiv \rho_0(r) \exp\left(-\frac{z^2}{2H^2}\right) \tag{2.3}$$

which also offers a quantitative definition of the previously introduced length scale H: it is identified as the standard deviation of the Gaussian vertical density profile. The following relation is also provided

$$H = \frac{c_s}{\Omega_K} \tag{2.4}$$

where Ω_K is the Keplerian angular velocity while c_s is the sound speed defined as $c_s^2 \equiv \frac{\mathrm{d}P}{\mathrm{d}\rho}$, which is equal to

$$c_s = \sqrt{\frac{k_B T}{\mu m_p}} \tag{2.5}$$

for an ideal fluid described by the equation of state 2.2.

The height scale H is thus proportional to $T^{1/2}$. Its radial dependence can be studied computing the radiative equilibrium of the disc, in order to obtain the radial temperature profile which is related to $H(r)$ through equations 2.4 and 2.5. Assuming a passive and flared disc, calculations lead to $T(r) \propto r^{-1/2}$ meaning $H(r) \propto r^{5/4}$ which is consistent with the "flared assumption", requiring the index of this last power law to be greater than 1. Figure 2.1b shows the vertical density distribution in a disc characterized by the just mentioned $H(r)$ radial profile and equation 2.3.

The overall dynamics of the gas is driven by the star accretion process. In order to explain this motion, some mechanisms of energy dissipation and angular momentum transport are needed. Viscosity plays a central role in this context. Detailed calculations show that shear viscosity, as modelled in an ideal gas and caused by molecular collisions is too weak to account for these processes. A proper model can be achieved assuming a highly turbulent regime that can produce a "turbulent viscosity" through the mixing of fluid elements at neighboring radii. The new parameter α called "Shakura-Sunyaev viscosity" which gathers all the ignorance on this process is introduced through the following reasoning: in the case of laminar flow, kinematic viscosity ν is defined as the constant of proportionality between the shear stress τ and the velocity gradient $\frac{\partial u}{\partial y}$, normalized with the fluid density ρ

$$\tau = \nu \cdot \rho \cdot \frac{\partial u}{\partial y} \tag{2.6}$$

The definition above is than generalized in its tensorial form to generic fluids. In an ideal gas the kinematic viscosity coefficient can be shown to satisfy $\nu = \frac{1}{3} c_s \lambda$, with λ indicating the mean free path of particles in the fluid. To describe the turbulent viscosity we proceed by analogy assuming $\nu_T \sim u_T \lambda_T$. In this equation u_T indicates a typical velocity of the turbulent regime which should satisfy $u_T \lesssim c_s$ because upper velocities would lead to shocks thermalizing the turbulent motion. The λ_T factor represents the typical length scale which, assuming isotropic turbulence, can not be greater than H, the disc height scale. Therefore, we obtain the equation

$$\nu = \alpha c_s H \tag{2.7}$$

which provides a method to estimate the turbulent viscosity introducing the α coefficient that, following the arguments above, must take values less than 1.

The dust component is modelled as a pressureless fluid with grains of different dimensions coupled with the gas medium. The strength of the coupling is expressed by the Stokes number St. Two drag forces come into play depending on the grain size s. If $s \lesssim \lambda$, with λ indicating the mean free path of gas molecules, the drag force is called Epstein drag. In this regime, which is usually the most relevant for most particle sizes, the drag is caused by gas molecules which collide against the front and back sides of the grain with different frequencies due to its motion relative to the gas medium. The Stokes number in the Epstein regime is related to the size and density of the particles that experience the drag. Once sizes much larger than the molecular mean free path are reached, the solid grains begin to experience a force of different nature called Stokes drag.

The model described above is a simplification of the complex dynamics of protoplanetary discs. There are many secondary effects which contributes to the proper description of discs evolution such as magnetorotational instability, photoevaporation and turbulences or vortices generated by the fluid motion in the turbulent regime. All this effects must be taken into account to obtain a complete description of the disc dynamics and evolution.

2.3 Observations

Direct observations of protoplanetary discs are of crucial importance for the detection and study of their morphology. Properties of substructures, such as their shape and dimension, had, in fact, been proved to reveal key features of the disc itself and of the objects that drive their formation. I am going to discuss how they are detected.

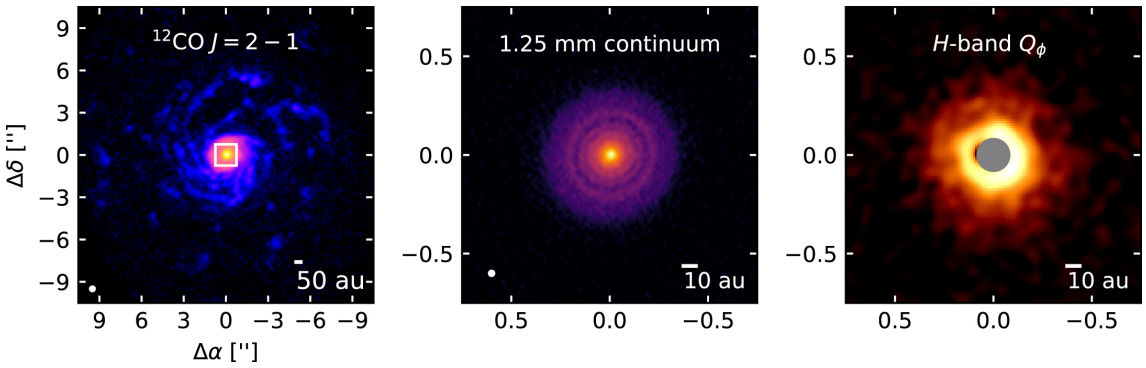

Figure 2.2: Comparison between observations of the RU Lup disc through the three main tracers. Image taken from ()

Due to their distance we can only look at their electromagnetic emission and exploit the natural diversity of structures and properties. Three sources of light can be identified.

The first one is the radiation, usually of micrometric wavelength, emitted by the host star and scattered by dust grains. This tracer is especially sensitive to the vertical structure of the disc allowing measures of the height and of the dust distribution along this direction.

The second tracer is the thermal continuum emission of solids, typically at millimetric wavelengths, emitted in the optically thin regime which implies direct proportionality between the intensity of the radiation observed and the mass density of the dust. Images of the disc thermal emission are, thus, of specific interest in the investigation of the solid grains.

On the contrary, the gas component eludes direct observations due to the nature of H_2 the dominant molecule. This part of the disc can be detected thanks to spectral line emission of less abundant molecules that constitute the third category of observational primers. Measures made from these probes suffer from the uncertainties in the molecular abundance of the revealed gases along with other problems which make gas observation complex and uncertain.

In all of these cases the range of wavelengths probed is approximately $1\,\mu\text{m} - 10\,\text{mm}$, hence going from infrared to radio waves. As anticipated, protoplanetary discs have a diameter of about 100 a.u. at a typical distance of 100/150 pc. Therefore, telescopes need to acquire an angular resolution under the arc second which is done using interferometric techniques. The leader observatory able to carry out this type of observations is the Atacama Large Millimiter Array (ALMA) which is an array of 66 high precision antennas that can be moved to achieve different configurations. Other facilities that have given a significant contribution to the detection of protoplanetary discs are the Very Large Telescope (VLT), the W. M. Keck Observatory and the NASA Hubble Space Telescope.

The images with the best resolution currently obtained are those of the DSHARP (Disk Substructures at High Angular Resolution Project) survey, ~ 0.035 arcsec, designed to examine properties of small-scale substructures and inquire how they relate to the planet formation process.

2.4 Planet formation

In 2006 the International Astronomical Union defined a planet as a celestial body in orbit around the sun with enough mass to reach a hydrostatic equilibrium (nearly round) shape and which has cleared the neighbourhood around its orbit. Outside the solar system the term planet indicates large bodies, in orbit around a star, with a mass below the limit for thermonuclear fusion of deuterium, currently calculated to be 13 Jupyter masses for objects of sun metallicity. The lower mass threshold should instead be the same of that considered for the solar system.

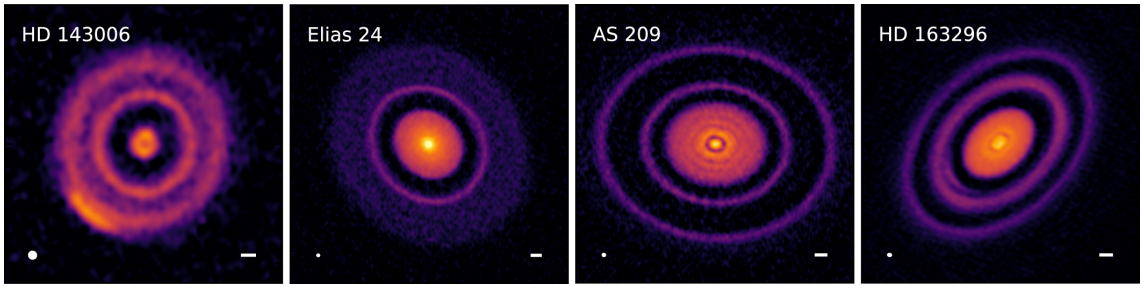

Figure 2.3: Examples of gaps and double gaps. Observations of the dust continuum emission of some discs obtained from DSHARP. Image adapted from ()

The study of extrasolar planets, often referred to as exoplanets, is of primary importance to make statistical studies of theories involving planets and their formation. Protoplanetary discs are the planets' birthplace. Different mechanisms of planet formation have been proposed and some parts of the theories are not completely understood yet.

The core accretion process is believed to be the origin of planets from the dust component of the disc. The process starts from the small dust grains which are well coupled with the gas. The coupling makes them acquire low relative velocities that result in gentle collisions forming aggregates called, when a significant mass is reached, planetesimals. While gradually accreting their mass, this aggregates settle towards the mid-plane and start to decouple from the gas experiencing a radial force which makes them spiral inwards toward the global maximum in the gas pressure. This motion is called radial drift and it is usually more effective for grains of sizes in the range 1-10 mm. The accretion process of planetesimals continues until the local dust population is depleted by radial migration or the collisions become destructive.

This process profoundly influences the morphology of the disc leaving us ways to detect and characterize newborn planets. Gaps, annular regions of dust or gas depletion, are the pivotal substructure generated from the interaction of the disc with young planets. Links between their shape and planets features were found, which will be discussed in detail in the next chapter: a measure of their width allows, for example, an indirect estimation of the planet mass.

2.5 Gaps

Gaps are going to be the object of application for the investigative approach proposed in this thesis due to their role in the detection of young planets. For this reason I am going to throw some light on their properties.

They are one of the most common substructure observed in images of protoplanetary discs, and planets are often the cause of their formation. However, other mechanisms were recognized to generate gaps including magnetorotational instability turbulence, gravitational instabilities, condensation of molecular species along different snowlines, large-scale vortices and self-induced dust pileups. The effective presence of a planet must hence be cautiously investigated and confirmed before inferring its features from the gaps properties.

The figure 2.4 presents the radial profile of the gas and dust densities at a gap. Taking into exam the graph relative to the dust density, I am going to highlight some recurrent features: at the core of the depletion zone, we can observe two minimums separated by a local maximum which usually coincides with the planet orbit. Sometimes this maximum reaches values comparable with the density of the unperturbed disc, a substructure that can be identified as a double gap. The borders also presents local maximums. Gaps are usually wider and deeper in the dust rather than in the gas component where they could even fail to form.

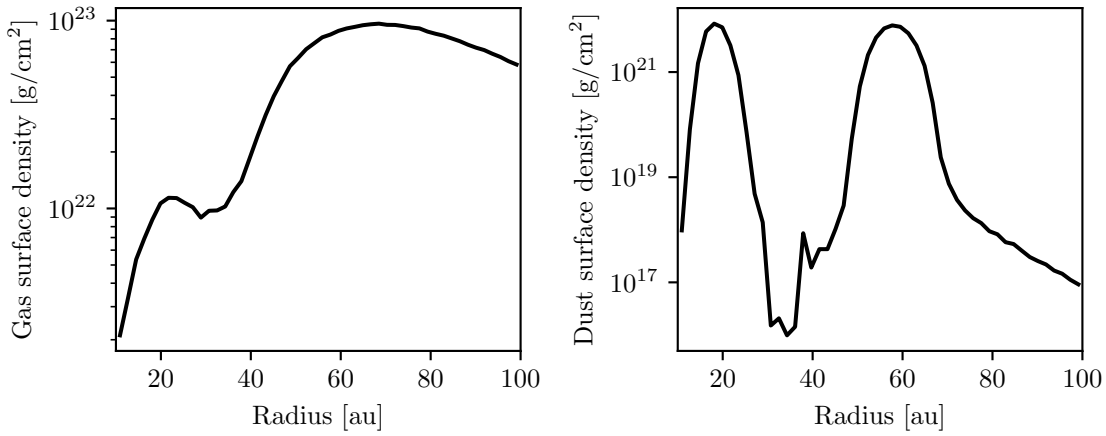

Figure 2.4: Azimuthally averaged radial profiles of the surface density of the gas component (left panel) and of the dust (right panel). The graphs were computed from a snapshot of a hydrodynamical simulation of a disc with an embedded planet.

Width and depths are the crucial properties of gaps used to investigate planet properties. Despite their importance there are not agreed on definitions. For the work done in this thesis they are not needed as the neural network model we aim to build would work directly with disc images. However, currently used methods, which will be discussed in the next chapter, include analytical tools which explicitly require the width value that thus needs to be defined. There are two main definitions used to give a value to the gap width: the first one defines it as the full width at half maximum of the radial profile density while the second one measures it as the radial distance between the minimum and the outer maximum. Due to the empirical nature of the analytical formulae that will be discussed, a change in the chosen definition requires to simply recompute some coefficients.

The depth is usually expressed as the minimum of the density profile normalized with the initial unperturbed value. Sometimes the intensity of the emitted light is used in place of the density to obtain a definition which depends upon directly measurable properties.

Chapter 3

State-of-the-art characterisation methodologies

Protoplanetary discs are interesting objects both in their own dynamic and in the context of planet formation. Many models were developed to describe and explain the disc motion and the origin of substructures and some questions are still open. However, these problems will not be the object of this chapter, here I am going to focus on the methods developed to interpret the disc images obtained through direct observations.

3.1 Addressed questions

The only property of protoplanetary discs that can be directly measured is the intensity of the light emitted with the different mechanisms previously discussed. The other features and parameters which characterize them and the embedded astronomical objects must be extrapolated from these measures. Tools and methods to achieve this purpose are central branches of research. In this chapter I am going to discuss the techniques so far developed and used, comparing their strengths and limitations in order to present the context in which the use of machine learning techniques will be proposed.

Features that can be extrapolated from the data fall into three categories: optical properties, hydrodynamical properties and properties of the central star and orbiting planets. Determining them from observations is crucial to properly apply the developed models, test theories and make predictions on the disc behaviour. These features are not all unrelated, in some cases equations obtained from the theoretical models can link some of them while for other ones some empirical relationships can be found. This intricate net of dependencies come useful in the characterisation of the disc and provides ways to check the results.

The following sections focus on the analysis of the techniques used to infer the mass of gap opening planets, which is the problem chosen in this thesis to investigate the machine learning approach. However, each method presented must be considered an example for the respective category. In fact, with proper adjustments, the same approaches are applied to infer other properties of the planet or of the disc.

3.2 Analytical formulae

Observations provide direct measures of the light tracers intensity, both spatially resolved and as the total disc luminosity, often expressed separately for wavelength or source. In addition to this data, from the images obtained, substructures' shapes and dimensions can be observed and

computed. A first attempt to interpret the data consists in researching empirical patterns in their relation with some interesting features of the disc. These links are often expressed as power laws or polynomial equations whose parameters are determined through a regression of available data. For example, the scaling relation between the disc luminosity at millimetric wavelengths and the star mass $L_{mm} \propto M_*^{1.7\pm0.3}$ is exhibited by discs with mean ages $\lesssim 3$ Myr. In addition to empirical formulae, theoretical models, supported by assumptions on the optical properties of the gas and dust in the disc, offer some equations useful in the determination of structural properties.

3.2.1 Planet mass and gap width

In the context of planet formation special attention is reserved to the study of gaps. We have seen that a possible explanation regarding their origin can lie in the interaction with forming planets. If we consider axisymmetric gaps, their width is the main discriminative property and has been linked to the mass of the planet responsible for their origin. For example, a direct proportionality between the gap width and the planet's Hill radius has been suggested, where $R_H = (\frac{M_P}{3M_*})^{1/3} R_0$ is the Hill radius, M_P the planet mass, M_* the stellar mass and R_0 the planet orbital radius. This leads to

$$M_P = (\frac{w_d}{k \cdot R_0})^3 \cdot 3M_* \tag{3.1}$$

providing a relation between the planet mass and the gap width w_d measured in the dust component. The coefficient k is introduced to model the proportionality and must be regressed from the data.

A different approach which had been proposed suggests the following relation

$$M_P = 0.0021 \cdot \left(\frac{w_g}{R_0}\right)^2 \cdot \left(\frac{h_0}{0.05}\right)^{\frac{3}{2}} \cdot \left(\frac{\alpha}{10^{-3}}\right)^{\frac{1}{2}} \cdot M_* \tag{3.2}$$

which links the planet mass to the gap width measured in the gas component w_g. This last equation includes more features related to the hydrodynamic of the disc which could, in principle, influence the development of different gap structures: the aspect ratio at the planet position h_0 and the α-viscosity. Accounting for these additional features should result in more accurate predictions across protoplanetary discs in which they differ significantly. However, measuring the α-viscosity or the aspect-ratio is not an easy task, and it introduces a source of error. Another downside of this equation is that it uses the gap width w_g measured in the gas component whose density map is more difficult to observe and resolve.

3.2.2 Strengths and limitations

Analytical formulae provide a quick method to determine features, such as the planet mass, with very low computation. Furthermore, an analytical expression highlights relationships between variables which could be interpreted through theoretical arguments in order to achieve a better understanding of the underlying physics.

On the other end, analytical expressions, especially the empirical ones, suffer in accuracy due to the extreme simplification of the functional forms used, and highlight relationships between a restricted subset of the variables which characterize the disc, potentially hiding the role of other ones. Additionally, in order to obtain the constants that characterize these formulae we need a set of data with known values of every variable involved. This requires having other methods to obtain this information or, in alternative, resort to numerical simulations which start from known values of the parameters and produce data analogous to the observations of real discs. In this case also the limitations of the numerical approach must be taken into account.

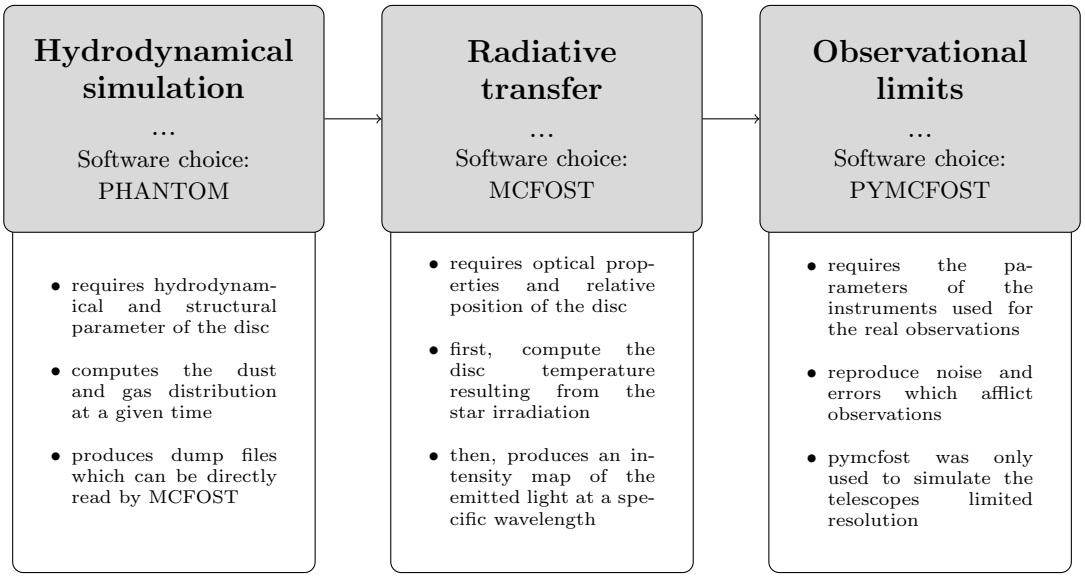

Figure 3.1: Typical workflow of a numerical simulation aimed at reproducing a synthetic image of a protoplanetary disc

3.3 Numerical approach

Computer simulations apply the models developed to describe the physics of protoplanetary discs and return predictions in the data space, which can then be directly compared with observations. The remarkable computational power that we can nowadays achieve allows the application of more complex models which reduce the approximations and are thus able to obtain more accurate results.

This approach can be used to infer discs or planets features, exploring some values through different simulations whose output is then compared with the actual data to decide the best fit. The simulation workflow (Figure 3.1) consists in three steps. First, starting from an appropriately chosen initial condition, we compute the fluidodynamical evolution of the disc obtaining its configuration at a given time. The second step is aimed at determining the radiative emission and trace it to simulate an observation from a specific distance and relative orientation. In order to predict the distribution and intensity of the light emitted, this step begins computing how the star radiates its energy and producing a map of the disc temperature. These processes involving the transfer of energy in the form of electromagnetic radiation are grouped under the term "radiative transfer". Finally, the last step has the purpose to reproduce the noise and limited resolution of the instruments used to collect the real data.

I am now going to give some insights for each of these steps. The images collected in the database for machine learning applications that we designed, were obtained through simulations of this kind. Among the possible choices of software available for each step, I am going to focus on the ones used to obtain the synthetic images in this database.

3.3.1 Hydrodynamical simulations

Hydrodynamical simulations solve fluid dynamics equations to predict the evolution of the gas and dust modelled as fluids with the specific features discussed in section 2.2. To run the models a starting configuration must be provided which is usually generated from given masses of each component and specifying, assuming axisymmetric discs, the density and sound-speed radial profiles

modelled with power laws such as:

$$\Sigma(r) = \Sigma(R_{ref}) \cdot \left(\frac{r}{R_{ref}}\right)^{-p} \tag{3.3}$$

and

$$c_s(r) = c_s(R_{ref}) \cdot \left(\frac{r}{R_{ref}}\right)^{-q} \tag{3.4}$$

where R_{ref} is a reference radius. The density and sound-speed values at this radius, in addition to the indexes p and q are the input constants that characterize the initial state. Other variables, such as the α-viscosity, the planet and stellar masses and the orbital radius, are also provided as input parameters.

One of the available programs for this step is PHANTOM, a smoothed particle hydrodynamics (SPH) code specifically designed for astrophysical applications and which is the one used to produce the starting data for the synthetic images collected in the database that was developed in this thesis. If properly compiled, PHANTOM is able to run a simulation of a protoplanetary disc with a planet orbiting around the central star. The simulation starts from an initial condition described in the input files and produces, after each given number of completed planet orbits, a dump file containing the spatial distribution of the gas and dust simulated particles, in addition to their velocity. All the information needed to restart the simulation is also stored inside these files.

3.3.2 Radiative transfer

The next stage in the simulation workflow is the radiative transfer computation. For this step we chose to use MCFOST, a code designed for this purpose and based on Monte Carlo methods. This program provides the possibility to run a simulation for gas and dust grains in the specific spatial configurations stored in a PHANTOM dump file.

The aim is to obtain a synthetic observation of specific light tracers as seen from given angles and distance. To produce the images used in this thesis we are only interested in probing the dust component through the thermally emitted and scattered radiation at a fixed wavelength. First of all, the radiative transfer of energy from the star to the disc elements is calculated, generating a map of the disc temperature. At this point the thermal emission of the dust is computed and ray traced to obtain the resulting image.

The model run by MCFOST depends upon optical properties, such as the optical length and the opacity, which are new parameters characterizing the results of the entire simulation process.

The output obtained at this stage is a 1024x1024 pixels image in the FITS (Flexible Image Transport System) format. The header of this file stores some input parameters: the wavelength of the light in the image, the inclination and position angle of the disc, its distance and the units of the data stored.

3.3.3 Generation of synthetic images

The real observations of protoplanetary discs are, being experimental data, susceptible to statistical and systematical errors and limited by the resolution of the instruments. The last step of the simulation process consists in reproducing these limits in order to obtain images as close as possible to those that we would obtain from a real observation. Among the different types of errors, the tools available reproduce the statistical fluctuations while, systematical errors are usually not introduced as they should be looked for and removed or corrected from the original measurements. This step is needed to allow a better comparison with the real images or to train machine learning models which should then be able to work with the real observations.

There are different tools which could be used to achieve this purpose. The CASA (Common Astronomy Software Applications) package is the most complete choice which is specifically designed

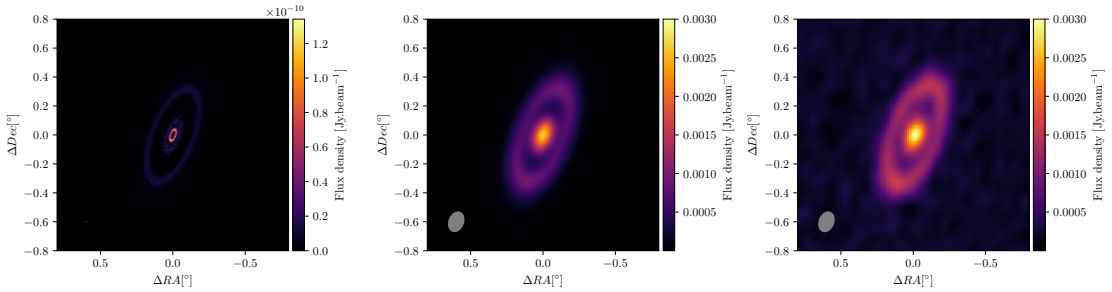

Figure 3.2: Comparison of a real disc observation, in the dust continuum emission, with synthetic images computed with the purpose of reproducing the real one. Left: image obtained as a result of a radiative transfer simulation through MCFOST, center: the same image at its left processed with pymcfost to reproduce the limited resolution, right: the real image. Add sources...

to reproduce images as they would be captured by ALMA, reproducing the noise in the data, the limited resolution of the instruments and taking into account the spatial configuration of the radio telescopes and other settings which could influence the observations.

The images in the database assembled in this thesis were processed using `pymcfost`, a less powerful python package designed to work with the files produced by MCFOST. I used this software only to reproduce the telescope limited resolution while statistical errors were not introduced. The method used is the convolution of the MCFOST image with a two-dimensional Gaussian parametrized by its full widths at half maximum called beam sizes. The final result of this process is still a 1024x1024 pixels FITS image.

3.3.4 Strengths and limitations

Computer simulations present numerous strengths which make them the preferred method for characterizing protoplanetary discs. First, the models simulated can be complex and describing the physics at a low level of abstraction allowing to account for secondary effects and unexpected interactions which would otherwise been neglected. They, furthermore, have a wide range of applicability as can be used, comparing the results with observations, to search or confirm values of most of the physical parameters characterizing the disc. The comparisons can be evaluated through proper metrics that quantify the accuracy of the initial ansatz. Unlike the analytical formulae, in this method the whole disc image and thus, in principle, the whole set of the disc physical features are involved in the predictions that are being made.

However, the power and flexibility of the computational approach comes at some costs. First, the fact that the whole disc modellization concur to the results might hide the existence of simpler and more interesting links between some parameters. The major downside is yet the high computational cost of a simulation which can run for hours or days. To properly investigate a parameter, the process which is usually applied consists in executing multiple simulations varying its value and finally comparing the results with observative data. The need of multiple simulations makes the time needed rise up to weeks.

Last, a problem, which is worth mentioning since it affects all the investigative methods, is that of degeneracies afflicting disc appearances at specific wavelengths. This means that different sets of parameters may result in the same disc structure and emission. They might be caused by the simplifications made in the model or be intrinsically due to the disc physics. In the latter case the same degeneracies would also affect real observations.

Chapter 4

Machine learning and neural networks

The term machine learning was introduced by Arthur Lee Samuel in 1959 with his research on the subject applied to the game of checkers. In 1998, Tom Michael Mitchell defined machine learning as the study of computer algorithms that can improve automatically through experience and by the use of data. This collection of tools and methods is usually considered a branch of artificial intelligence and finds its natural application in pattern recognition and data analysis tasks for which is arduous or unfeasible the development of conventional algorithms. The past decades have shown an incredible development in this field, with applications in a wide variety of subjects.

In this chapter I am going to introduce a specific machine learning tool, neural networks, explaining their key properties and pondering their application to the open issues, in the study of planet formation, discussed in the previous chapter. I am also going to review a previous attempt at the use of this approach in the analysis of protoplanetary discs.

4.1 Neural networks

Artificial Neural Networks (ANN) are a group of machine learning algorithms inspired by the structure and functioning of biological brains. They consist in networks of artificial neurons, which are the elementary units capable of receiving input signals, processing them and activating their output accordingly. The connections between the neurons account for the complexity of neural networks making them an extremely manifold and versatile tool: the overall architecture and additional parameters can be tuned to design models specifically suited for the problems addressed. All these elements will be explained in the upcoming sections.

Across their different applications, the two main categories of problems they are commonly used for, are classification and regression. In the first case, the neural network labels each input assigning it to one of a previously defined set of categories. A neural network that detects which animal is depicted in a given image is an example of a classifier. In the second category of problems, the neural network is instead used to infer, from the given data, a relation between some variables. The models presented in the next sections are all examples of regressors.

4.1.1 The artificial neuron

The elementary units of neural networks, artificial neurons, are mathematical objects which model the structure in figure 4.1: they receive a set of input values and process them returning a single output signal. Essentially they are parametric functions $\nu(\vec{\mathbf{a}}) : \mathbb{R}^k \to \mathbb{R}$ where k indicates the

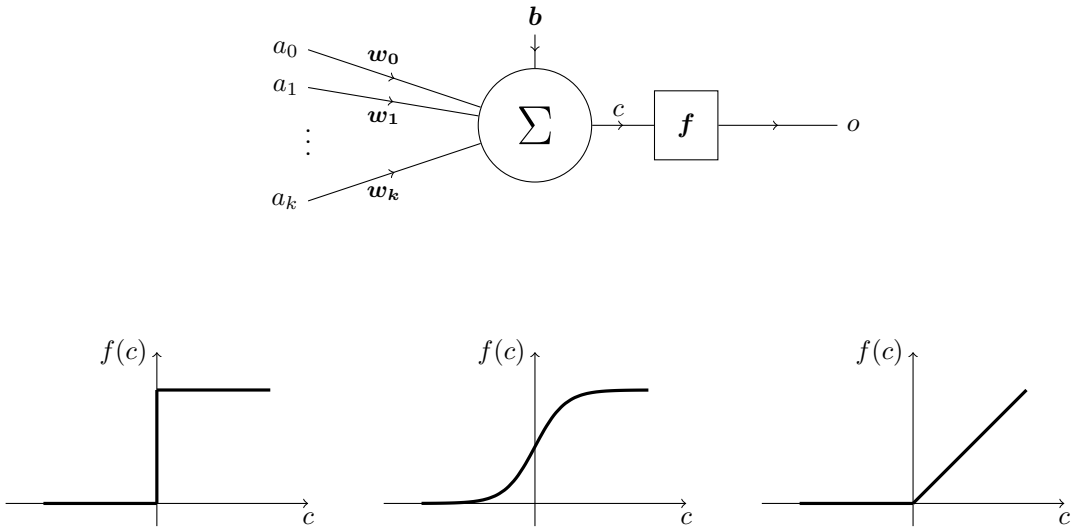

Figure 4.1: Top panel: scheme of an artificial neuron. Bottom panel: graphs of three possible activation functions. From left to right: Heaviside step function, sigmoid and ReLU

dimension of the vector $\vec{\mathbf{a}}$, equal to the number of input connections. To explore the arguments behind their design and unfold the comparison with the biological unit, this relation can be rewritten as the composition of two functions corresponding to the two blocks in figure:

$$\nu(\vec{\mathbf{a}}) = g \circ \Sigma = g(\Sigma(\vec{\mathbf{a}})), \quad \begin{array}{l} \Sigma : \mathbb{R}^k \to \mathbb{R} \\ g : \mathbb{R} \to \mathbb{R} \end{array} \tag{4.1}$$

The first block, Σ, performs a weighted sum of the input values a_j introducing also a bias b

$$c = b + \sum_j w_j \cdot a_j \tag{4.2}$$

where the j index identifies the input connections of the artificial neuron considered. The weights w_j and the bias constitute the set of parameters in which lies the potential of neural networks. This first process mimics the ability of biological neurons to adjust the intensity of the signals in its connections modifying the ratio of neurotransmitters and inhibitors. Still in the biological counterpart, the total potential obtained integrating the input signals is then transmitted to the connected neurons if it reaches a certain threshold. The same behaviour is modelled in artificial neurons in the second block, with a function $f(c)$, called activation function. It takes the c value obtained in the previous step and returns the output $o = f(c)$ which is then transmitted to the connected neurons becoming one of their input values.

In the choice of the function f, its role as an activation threshold should be kept in mind. The bottom panel of figure 4.1 shows three examples of functions that could be chosen for this purpose. The first and simpler is the Heaviside step function which returns 1 if the input value is positive and 0 otherwise, simulating the discrete behaviour of biological neurons. An improvement to this choice, showed at its right, can be achieved using the sigmoid

$$o = f(c) = \frac{1}{1 + e^{-k \cdot c}} \tag{4.3}$$

which unlike the step function is continuous and infinitely differentiable in 0.

Last, the third panel depict the ReLU (Rectified Linear Unit) activation function which returns the input value if positive and 0 otherwise. This is the most popular among the whole set of activation

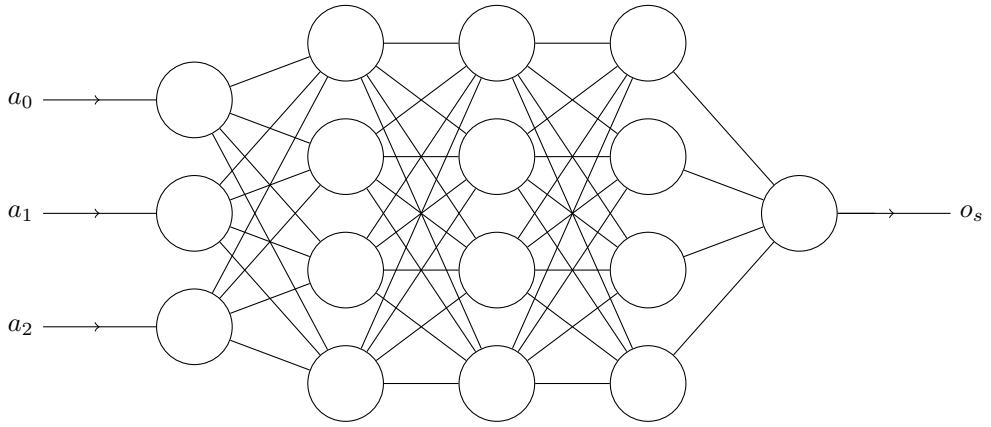

Figure 4.2: Scheme of a feedforward neural network

functions available, due to its simplicity, requiring very little computation, and some mathematical properties that make the training process using gradient descent methods particularly efficient. Moreover, this function is non-linear, not limited and can return any real positive values. The replacement of discreet outputs with a value varying continuously is still supported by the biological comparison. In fact, even if real neurons can only send discreet signals deciding whether or not to activate their output, the frequency of transmission might vary. The continuous output thus model this possible behaviour.

4.1.2 Architecture and types

We have thus seen that artificial neurons, the building blocks of neural networks, are fundamentally simple parametric functions. Just as in biological brains, the complexity and flexibility of neural networks is due to the number and nature of neurons connections. An enormous collection of different architectures was developed since the advent of the field, some of them are specifically designed for a certain task while others can be applied to a wider set of problems with specific strengths and limits.

The simplest and most used are called feedforward neural networks. Their structure can be decomposed in set of neurons, without any connection between them, called layers which are organized sequentially in a specific order. The first and last layers are called respectively input and output layer because of their function in the network, they provide the interface for sending and reading data to and from the model. The other internal layers, without connections with external resources, are called hidden layers. Neural networks with 2 or more hidden layers are referred to as deep neural networks and their application constitutes the subject of deep learning. The neurons in each layer receive, through their input connections, the outputs produced by the neurons of the previous layer. The information thus moves in only one direction: forward. In a "dense" network all the neurons of two consecutive layers are connected.

Maintaining the formalism previously introduced, this means that in a dense neural network the n^{th} layer process the vector of input values $a_i^{(n)}$ coming from the $(n-1)^{\text{th}}$ layer producing the output vector $o_j^{(n)}$ through

$$o_j^{(n)} = b_j^{(n)} + \sum_i w_{ji}^{(n)} \cdot a_i^{(n)} \tag{4.4}$$

where $w_{ij}^{(n)}$ is the matrix containing the weights of all the connections between the current and previous layer. The whole neural network is thus equivalent to a complex parametric function $N(\vec{\mathbf{a}}) : \mathbb{R}^m \to \mathbb{R}^h$ where m is the number of input variables while h is the number of values

returned:

$$N(\vec{\mathbf{a}}) = o_s = b_s^{(n)} + w_{sr}^{(n)} \cdot f(... \cdot f(b_k^{(2)} + w_{ki}^{(2)} \cdot f(b_i^{(1)} + w_{ij}^{(1)} a^j))...) \tag{4.5}$$

The weight matrices and the bias vectors are the parameters that can modify the functional form.

For the specific problem addressed in this thesis, convolutional neural networks could be a better alternative to the feedforward architecture. This type of networks is specifically designed for image analysis, it is similar to feedforward neural networks with the addition of specific layers that process the input data in order to achieve space invariance. With this addition, convolutional neural networks can be trained to detect patterns independently of their position, size or orientation inside the image resulting in more flexible and effective models. These layers typically consists in two-dimensional filters which slide along the input image, usually of greater dimensions, producing a response called "feature map" that is then sent forward to the next layer. In addition to the achievement of translational equivariance, a convolutional neural network applied to images, potentially of large dimensions, reduce the number of trainable parameters if compared to a regular neural network designed to work with the same images. This simplification improves the performance of the model and helps avoid problems, such as overlearning, during the training process.

4.1.3 Training

In the context of machine learning, training algorithms are usually classified in three categories: supervised, semi-supervised and unsupervised. As the name suggests, supervised machine learning algorithms use a set of pre-labelled data to train the model, comparing the results with the known correct answers and adjusting the parameters accordingly. Instead, unsupervised machine learning models implement algorithms that are able to learn directly from unlabeled data. This can be achieved, for example, with the introduction of a feedback mechanism that rewards the best solutions. Methods which fuse parts from each approach are classified as semi-supervised. I am going to focus on the former which is the one suggested and used in this thesis.

In order to train the model, a tool to evaluate its performance is necessary. The "loss function" is introduced for this purpose. In the case of neural networks built for regressions, the mean squared error (MSE) of the predicted value is usually used. However, other choices are possible.

We have already seen that neural networks are essentially complex parametric functions which could potentially approximate any functional relation between the input and the output space by varying the weights of the connections. The evaluation of this model with the loss function can hence be abstracted as a function from the space of the network's weights to $\mathbb{R}$:

$$\mathcal{L}(\vec{\mathbf{w}}) : \mathbb{R}^k \to \mathbb{R} \tag{4.6}$$

where k is the total number of trainable parameters in the model. At this point, it is clear that training a neural network is simply an optimization problem. The aim is to find the set of parameters which minimizes the loss function evaluated using the training dataset.

For this process numerous algorithms, called optimizers, were developed, revolving all around the method of gradient descent which is a technique based on the fact that the gradient of a multidimensional function, such as (4.6), computed at a specific point, indicates the direction of steepest ascent. To exploit this property, the process starts from a set of initial weights which is a point in the parameters space, then the gradient of the function to optimize is computed, and the parameters changed accordingly to reach the minimum. After each step t, the weights of the neural network are thus updated through

$$\vec{\mathbf{w}}_t = \vec{\mathbf{w}}_{t-1} - \eta \vec{\nabla} \cdot \mathcal{L}(\vec{\mathbf{w}}_{t-1}) \tag{4.7}$$

where the hyperparameter η called learning rate was introduced. As the name suggests, it determines the length of each step of the walk in the parameters space towards the minimum, varying,

as a consequence, the speed of the process. The entire algorithm is divided in epochs, each of them consists in an update of $\vec{\mathbf{w}}_t$ using all the data in the training set.

Finally, the computation of the gradient is itself a challenging task. The algorithm used for this purpose in the context of neural networks is called back propagation.

The steps unfolded here are the skeleton of almost every optimizer developed for the training of artificial neural networks. They then differ in some additions made to this main algorithm to achieve better performances and increase stability. For example, in some optimizers, the equation 4.7 is modified adding a term called momentum which is computed considering the previous steps and thus adding history to the algorithm. The name derives from the analogy with physical momentum. This addition is designed to accelerate the optimization process and improve its capability.

4.1.4 Training data

The training process described requires a large set of data from which the neural network could be trained. To implement supervised learning algorithms, this data has to be labeled, which means that for each item representing a possible input, also the respective expected output has to be provided.

The dimension and quality of this dataset deeply influence the results obtainable from the training process. In order to improve the results and achieve a successful model this dataset should contain numerous items with varying features representing the entire hypervolume of allowed values for the variables characterizing the input data. There is not a proper rule to establish how many items should be contained in this dataset, generally they should be as many as possible, while the minimum required depends upon the targeted accuracy and the complexity of the chosen model to train.

4.1.5 Hyperparameters

The training is an important step in the process of acquiring a successful model but, even having access to a large and exhaustive dataset, previous choices regarding the building structure of the neural network might undermine the possibility to reach satisfying results.

The parameters whose value is set during the definition of the model are called hyperparameters. They include the type and number of layers, the number of artificial neurons for each layer and the choice of the activation function and of the optimizer, with its parameters such as the learning rate. In these variables lies the plasticity of neural networks as they can be accurately tuned to build a model with the potentiality to address the targeted problem.

Understanding how modifying specific features alters the behaviour of the neural network might help in this tuning process pointing towards the right direction. However, most of the time there are not precise prescriptions regarding the choice of these parameters in relation to the addressed problem and they are often set by trial and error.

More sophisticated techniques and algorithms had also been developed to automate this process and assure a better coverage of the hyperparameters space.

4.1.6 Overlearning and underlearning

The training process of a neural network could end in three different ways. The figure () shows an example of application to the same data of three different models. The panel in the middle represents the desired result: the neural network has been able to predict the functional form which originated the data without learning its statistical noise. The panels on the sides depict instead the two issues that might occur.

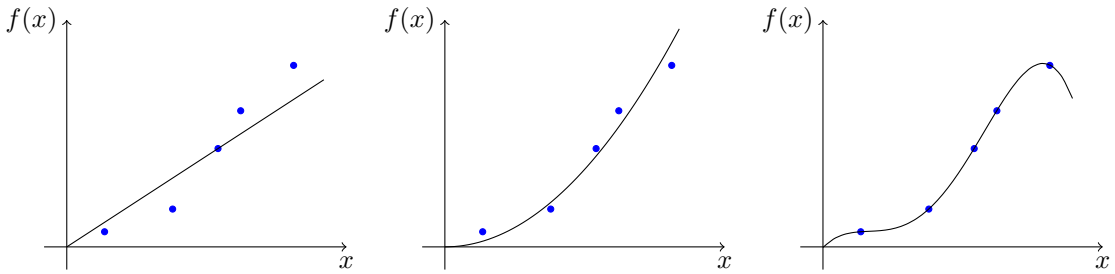

Figure 4.3: Three different neural networks applied to a regression problem. Left panel: underlearning, middle panel: successful regression, right panel: overlearning

The situation in the left panel is called underlearning. In this case the function inferred with the model oversimplifies the correct underlying relation failing to predict some of its characterizing features. The loss function and other metrics should return values indicative of the neural network bad performance, providing a way to detect this situation. To solve this problem, as a first attempt the number of epochs in the training process is usually increased. If a longer training does not improve the performances, the reason behind underlearning could be the inadequacy of the chosen neural network architecture. Improvements can hence be obtained adding more layers or neurons, and thus raising the complexity of the model.

The situation depicted in the right panel is instead called overlearning. In this case the problem is essentially the opposite: the model learned the training data with its noise inferring a complicated functional form that performs well on the training set but fails when tested with new data. In order to detect this problem, the dataset built to train the model is divided in two subsets. The first one, the training set, is used to train the neural network while the second, smaller, one, called test set, is used in a second phase to test the obtained model. The training set is usually assembled with about the 80% of the items available in the dataset. Comparing the loss function values obtained with the train and test datasets we can detect the overlearning. The possible solutions applied in this case are the exact opposite of that in the previous situation: we can reduce the number of epochs "early stopping" the training process or decrease the complexity of the model.

4.2 Strengths and limitations

Neural networks have been applied to a huge variety of problems spanning completely different fields. This flexibility is one of the main strengths of this class of machine learning techniques. The universal approximation theorem has shown that a finite number of hidden units can approximate any continuous function to any desired degree of accuracy. In addition, this tool has great tolerance to errors and noise in the data, and generalizes easily to new inputs.

In order to address a specific question with a neural network two elements are necessary: a large set of data for the training process and a model for the problem that allows the encoding of the input and output data in numeric values readable and computable by the neural network.

At one end, these features constitutes the strengths of this tool: as a consequence of the data-driven approach, no prior knowledge of the studied processes is needed and any existing relation can be unraveled through an enough complex model. On the other end, the need for a large enough database is usually the main downside, which complicates the application of machine learning techniques in some areas where collecting data, or labelling them with other methods, might result difficult. Furthermore, the behaviour of the trained model is unexplainable and possible simple analytical relationships between the involved variables remain hidden. Hence, the use of this tool does not provide new knowledge on the studied phenomenons. In a minor set of cases encoding the problem in a model suitable for a neural network might also result challenging.

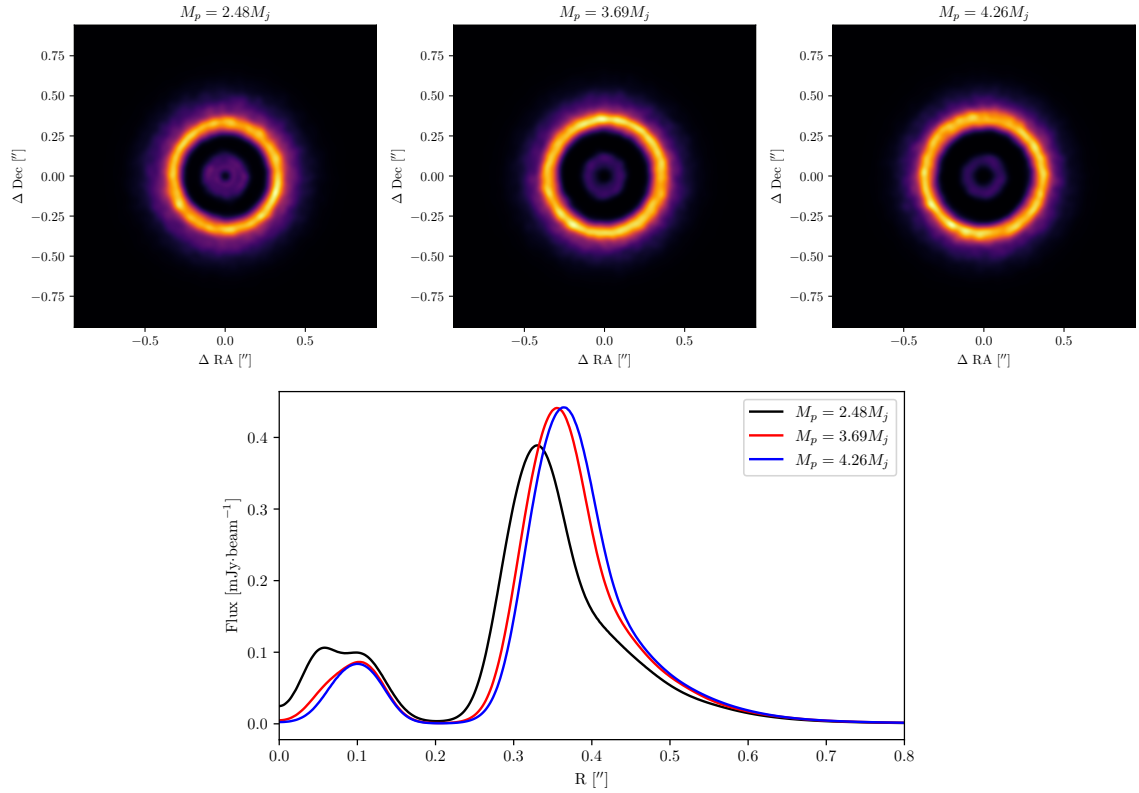

Figure 4.4: Three simulations of the same disc with planets of different masses. Top panel: map of the flux density of the dust emission at 1.3 mm, bottom panel: radial profile of the azimuthally averaged flux

Regarding computational complexity, depending on their architecture, neural networks might require a large amount of resources and time. They are however suited for parallel processing, even with GPUs, allowing an improvement of performances and a reduction of the time needed.

The most demanding part is the training process because of the back propagation algorithm and the repeated iterations. The forward propagation of the data can instead be performed very quickly. This is an advantage since the training process should be run only once to eventually deploy the trained model and use it to analyze new data. Compared to numerical simulations, in the specific case of hydrodynamical models, neural networks are more convenient in terms of computational cost.

4.3 Machine learning and protoplanetary discs

Machine learning algorithms, and specifically neural networks, had been successfully used in image analysis tasks. The flexibility along with the strengths in terms of computational cost and resistance to errors, makes them ideal candidates for use in the study of protoplanetary discs.

In this section, I am going to propose this new approach explaining the expected improvements and suggesting possible solutions to the critical points. I will then review a recent paper in which the authors used successfully similar methods to infer the mass of planets embedded in protoplanetary discs. I will hence highlight the differences with our approach explaining the reasons behind our choices.

4.3.1 The proposed approach

In the third chapter we have discussed the current methods adopted to interpret discs observations. Among them, numerical simulations are preferred, as they provide images directly comparable with observations in the data space and are more accurate than the analytical expressions available. In order to run a simulation some parameters characterizing the disc dynamics and its optical properties are needed. They have hence to be inferred from observations or, otherwise, included in the set of variables which will be varied across different simulations and thus determined selecting the best fit to data. However, the high computational cost of hydrodynamical and radiative computations limits the exploration of the parameters' space.

A machine learning approach could overcome these restraints. We propose the development of a neural network designed to read 2D images of light intensity at specific wavelengths, which can be produced from interferometric measures, and return the inferred value of one or more parameters characterizing that disc. Any feature related to the disc emission or the formation of visible substructures has the potential of being inferrable using this technique.

Among the possible applications, I will focus on the already mentioned problem of detecting young embedded planets and measuring their mass. Nevertheless, the same techniques could be easily generalised to other issues. In chapter 3 I mentioned some known analytical relationships which prove an existing underlying link between the planet mass and gaps morphologies. Picture 4.? shows the synthetic images obtained simulating three identical protoplanetary discs whose only difference resides in the mass of the orbiting planets. Even without any specific measure we can appreciate the different features of their substructures. All these evidences suggest that a regression through a neural network might be possible.

In our perspective the trained model should be able to analyze an image, obtained from direct observation of a protoplanetary disc presenting substructures due to forming planets, and provide measurements of the planet's properties. In order for the neural network to perform accurate predictions, it has to be trained with sundry images of discs with features properly varied to explore at best the ranges of their possible values. The training dataset can be produced collecting synthetic images generated from numerical simulations. With this method all the physical properties of protoplanetary discs and their embedded planets are manually set, allowing the control of the distribution of the values explored and assuring their correctness. Once the neural network has been trained, its application to a new input requires a few seconds which is a radical improvement if compared to the time needed by the numerical approach.

It could be argued that since the training dataset is constructed from the results of hydrodynamical and radiative simulations, the computational resources exploited, and does the time needed, are analogous to the use of the computational approach which, thus, could have been used to directly infer the desired features from the best fit. However, while this is true, we could achieve significative advantages with two simple strategies. First, we exploit the results of simulations already run by different people and research groups, in the past years, for sundry independent studies. This allows the implementation of a large and diversified dataset which would have otherwise required months and significant resources to produce the data from the ground up. Further improvements could also be obtained opening to the research community the possibility to contribute uploading their own images. Second, the neural network trained with a sufficiently exhaustive and representative dataset would be able to successfully analyze different images without having to iterate the training phase. If applied to a large pool of observations, the overall balance of the employed resources would thus favour the machine learning approach. Additionally, the evaluation of each image is fast enough to allow the use of this tool in hypothetical large surveys of protoplanetary discs to detect the presence of planets and infer their mass, obtaining a quick classification of the observed images. In this case, depending on the size of the survey, the computational approach might even be unfeasible while the analytical relations available are less accurate.

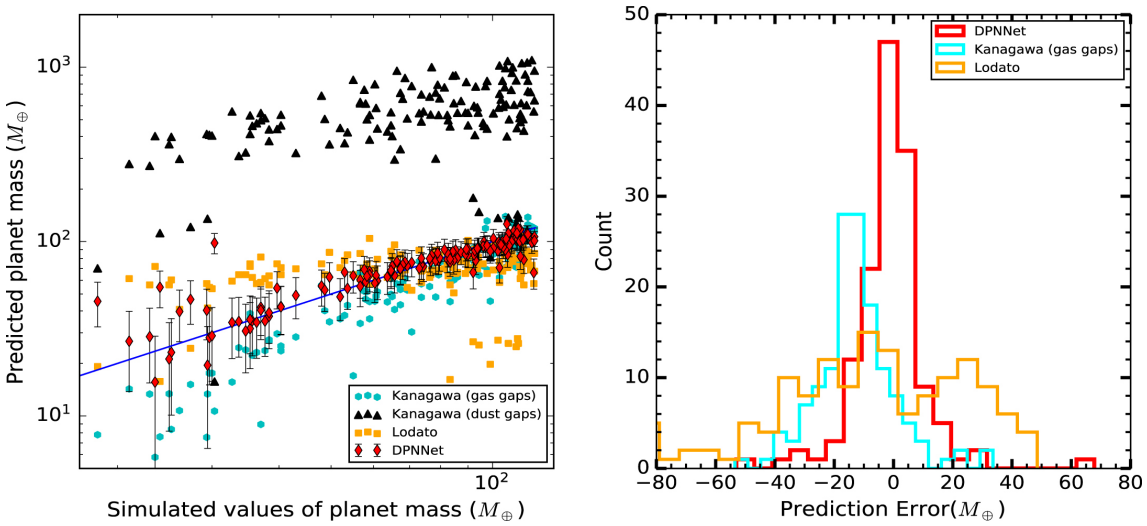

Figure 4.5: Figure 7 of Auddy and Lin paper

4.3.2 Previous attempts in literature

Some machine learning applications in this field have been already tried, with an approach similar to the one proposed. For example, in [insert reference] Min Kai Lin and Sayantan Auddy realised a training dataset with [insert number] two-dimensional simulations varying, in addtion to the mass of the embedded planet, the values of the α-viscosity, the aspect-ratio, the Stokes number, the gas to dust ratio and the index of the power law used to model the initial density. They then measured, in the obtained results, the width of the gaps formed in the dust component due to its interaction with the orbiting planet. A neural network was thus trained to infer the planet mass from this last measure and the other parameters previously mentioned.

The results achieved were promising. They compared the mass value predicted by the neural network with the results obtained applying the analytical formulae () and (). In the picture () we can observe this comparison. The neural network performed better than both the analytical approaches: its prediction error follows a normal distribution whose mean value and standard deviation are both smaller showing its better accuracy and precision.

The approach we propose in this thesis present substantial differences. First, we aim at reducing at its minimum the number of disc features needed and not directly obtainable, with low uncertainty, from observations. For this reason, we opt for a neural network designed to receive the entire observed image as input which also allows, as a fortunate consequence, to consider the entire morphology of the disc substructures.

Additionally, they considered only 2D simulations to reduce the computational power needed while we built our dataset with 3-dimensional results which allowed us to account for additional parameters, such as the different inclinations of the disc. We also included in the training set, images of the same simulation produced at different times, thus introducing an additional variable.

Chapter 5

Dataset design

Every machine learning application depends heavily upon the training dataset. Its realization is the most crucial aspect of the process and, in addition to collecting the data, involves designing how to organize and store them and checking their quality, ensuring that the entire parameter space is properly represented and that no bias was introduced during the production and selection stages.

In the approach we proposed, the aim is to build a database of synthetic images of protoplanetary discs probed through thermal dust emission, produced exploiting the results of already run simulations. The design and implementation of this dataset is the heart of this thesis. In this chapter I am going to present its content and the structure developed to organize and store the information. The scripts produced to support the use and maintenance of the database will also be explained, providing an overview of their functions.

5.1 The data

As I already mentioned, the starting data from which we built the database we designed to train neural network models are results of hydrodynamical and radiative simulations. At this stage of the simulation process, the resulting images are spatially resolved maps of the flux density of light at a specific wavelength as seen from a given relative position. This fictitious images are further elaborated with pymcfost, whose functioning has been discussed in chapter 3, to simulate the limited resolution of the telescopes through a convolution with a Gaussian beam. The final product is a set of images in the FITS format. In the next section I will explain how they are organized and indexed, while here I am going to discuss the content of each image providing insights on the source and possible usages of each parameter included. I will also present some global features of the dataset.

The FITS format encode the stored information in Header and Data Units (HDUs). Each one, as the name suggests, consists in an ASCII text header organized in key-values tuples, followed by a binary, usually multidimensional, array where the data is stored.

In our dataset, each FITS file contains only one image corresponding to a single HDU. In the header, some parameters of the disc images depicted are stored. The preprocessing step, run on the files obtained from the radiative transfer simulations, ensures the proper standardization of the header, retrieving and adding the missing variables. Table 5.1 shows the list of the header keys corresponding to each of these parameters saved in the FITS files. Brief descptions are also provided.

In this table, I divided the list in six different groups. The first one contains variables which are read from the PHANTOM's dump file and are thus hydrodynamical features of the disc. The next

KEY	content
RIN	inner radius of the hydrodynamical simulation domain
ROUT	outer radius of the hydrodynamical simulation domain
RC	reference radius for the exponential taper in the radial profile density
MGAS	total mass of the gas component
MDUST	total mass of the dust
HRIN	aspect ratio at RIN
RHOG	intrinsic grain density
PINDEX	power law index for the initial surface density profiles
QINDEX	power law index for the initial sound speed profile
TIME	time of the snapshot in complete orbits of the planet
ALPHASS	Shakura-Sunyaev α-viscosity
ECC	eccentricity of the planet's orbit
NSTARS	number of stars in the simulation
TSTAR#	temperature of the star (K)
MSTAR#	mass of the star in solar masses
RSTAR#	radius of the star in solar radius
NPLANETS	nnumber of planets in the simulation
MPLANET#	planet's mass in Jupyter's masses
RORB#	radius of the planet's orbit
DISKPA	disc's position angle
INCL	disc's inclination
DISTANCE	disc's distance
WAVE	wavelenght of the radiation showed in the image
GDRATIO	gas-dust ratio used in the MCFOST simulation
BUNIT	units of the data in the image
CDELT1	x pixel scale (deg)
CDELT2	y pixel scale (deg)
BMAJ	major FWHM of the gaussian beam
BMIN	minor FWHM of the gaussian beam
BPA	beams's position angle

Table 5.1: keywords in the fits file

two groups list the physical properties of stars and planets. These groups contain the "multiple keys", those in the KEY# format, which appear more than once, substituting the # character with an index, in case of multiple stars or planets. Except for the star temperature which is a parameter of the radiative transfer simulation, also these variables are retrieved from the saved standard output of MCFOST which read these properties in the PHANTOM's dump file. The fourth set collects some input variables of the MCFOST's simulation. They set the relative position and orientation of the disc and the wavelength of the radiative emission probed. The GDRATIO variable contains the value of the gas to dust mass ratio set when running MCFOST which is simulated independently to the value of the same parameter used to obtain the respective PHANTOM dump. For this reason computing MGAS/MDUST gives a different result since these last parameters refer to the results of the PHANTOM simulation. This method is preferred due to the less time required to run a radiative transfer simulation with respect to a hydrodynamical one. The obtained results are valid in the assumption of negligible back reaction between gas and dust which is usually true for sufficiently large gas to dust ratios ($\gtrsim 10$).

Finally, the fifth group provides the units of the stored data while the last one contains the dimensions and orientation of the Gaussian beam used to convolve the image.

The images collected within the database in the version deployed with this thesis, come from two different sources. The first group of data, provided by Dr. Benedetta Veronesi (...), consists in numerous simulations, both hydrodynamical and radiative, of the DS Tau disc [2]. I additionally re-run the radiative transfer through MCFOST to obtain images of the disc at different inclinations. From this source, after the further processing through pymcfost, we obtained a total of 20370 images. The second group of data was provided by Dr. Enrico Ragusa (...) and consists in sets of snapshots obtained from hydrodynamical simulations from which I selected only the ones which showed the presence of gaps. I then simulated the radiative transfer and produced images with different gas to dust ratios and inclinations. From this source we eventually obtained 4350 images.

For each result of a radiative transfer simulation, I used pymcfost to produce 10 images obtained convolving circular Gaussian beams of different sizes (FWHM) logarithimically spaced between 0.01 and 1 arcsec. Figure 5.1 shows an example of the results obtained. These values were chosen to explore resolutions realistically achievable, as proved by for example the DSHARP project, and additionally explore the upper and lower limits. Very resolute images are included for applications where it might be preferable to remove instruments limitations or to test models in the perspective of future improvements in the observational techniques. On the other end the less resoluted images are convolved with beams up to about the same dimensions of the disc. This limit may be useful to test what information could be extrapolated from spatially unresolved observations which essentially provide only measures of the light intensity but can not reveal the disc substructures.

The final product is a database made of 24720 FITS images with dimensions of 1024x1024 pixels. In this version each image contains only one star and one planet and exhibit an induced gap. Among the parameters listed in Table 5.1 only some of them vary significatively across different simulations. Apart from the beam sizes whose distribution has already been discussed, they are, the gas and dust masses, the gas to dust ratio, the planet's mass and orbital radius and the time. Picture () shows the distribution of these parameters and the scatter plots showing the correlations between these variables. Some of them are easily interpretable, for example the gas and dust masses decrease with time accompanied by the planet's mass increase can be understood as consequence of the planet's accretion process. There is also a strong correlation between the planet's orbital radius and its mass which is due to the third Keplerian law.

Most of the discs were captured, in the images, from three different inclinations of approximately 0, 30, and 60 degrees. Furthermore, all of them were computed for the dust continuum emission at 1.3 mm wavelength. The planet's mass, which is the target feature of the possible application on which we focused, shows an acceptable distribution over a sufficiently large interval of values. The gas to dust ratio is another parameter which has been widely explored.

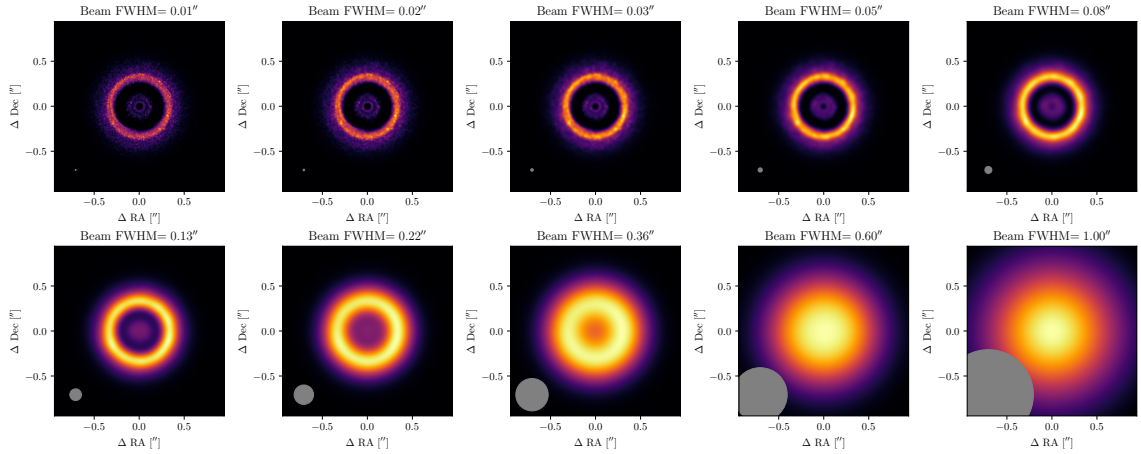

Figure 5.1: The ten different sizes of the gaussian beam used to convolve each synthetic image. In this figure all the images are obtained from the same simulation result varying only the beam size whichis shown in the left bottom corner.

5.2 Structure and interface

Every image presented in the previous section is saved as a different file whose name is in the format described by the regular expression

```
1          ^\d{6}_.{3,5}TIME\d+GD\d+W\d+B\d+.fits$
```

which provide some information about the parameters used to produced that image. The first six characters are numbers forming an index key which is used to uniquely identify each image. The collection is stored entirely inside a single directory where two additional files are generated: `index.js` and `data.js`. They are json files aimed at indexing all the information stored in the FITS files except for the images' data, in order to provide an easier access especially when the dataset is not completely available locally.

The entire dataset is, in fact, stored in a public github repository exploiting the git lfs storage system, and the typical usage workflow for which it was designed consists in the following steps. First, through a script provided, and explained in the next section, the index files can be downloaded. They can then be read to obtain a dictionary which is loaded in a pandas DataFrame containing a reference to each image and all the parameters stored in their headers. It is thus possible to analyze this data to select, in case, a specific subset of the entire database according to the implemented application. The keys of the images filtered in this way can thus be used to download them through the provided script, without the need to get the entire dataset whose size approaches hundreds of gigabytes.

To allow these functionalities, inside the index file, some additional parameters regarding each image are stored. They include the file names, their hashes and the unique keys associated to each item.

5.3 Supporting scripts

To populate and manage the database, I developed the python package `dblib` and some command line scripts which provide immediate access to their main features. The structure of the package is the following:

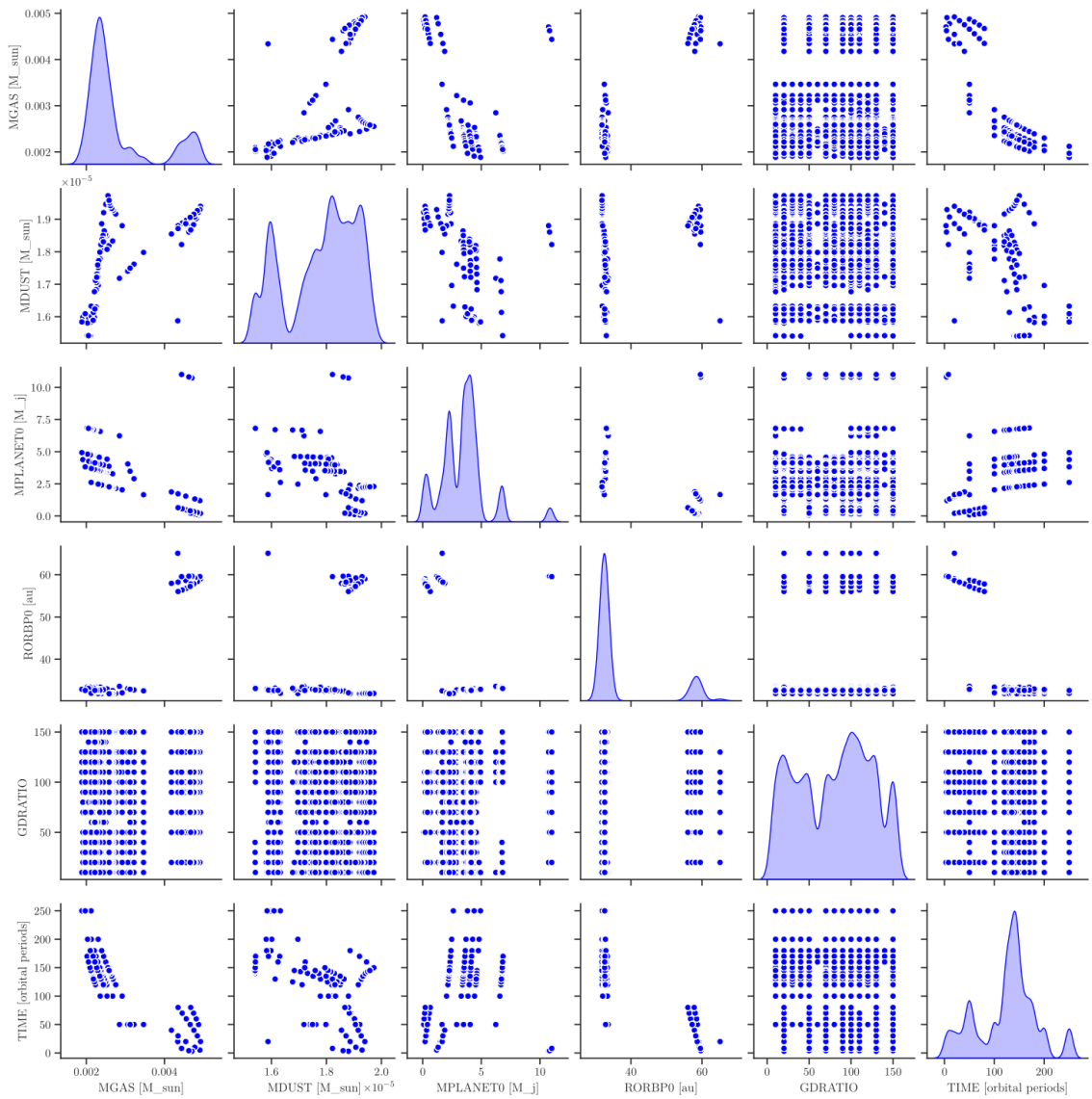

Figure 5.2: Pairwise relationships in the deployed dataset. The diagonal shows how each parameter is distributed while the off-diagonal plots highligth correlations among these variables. Only the parameters whose values vary significantly within the dataset are shown.

```
dblib
├── index_keys.py
├── index.py
├── pymcfost_process.py
├── utils.py
└── __init__.py
```

These modules fulfill three main functions: preprocessing through `pymcfost` and standardization of the images obtained after the MCFOST simulation, generating and accessing the index files and last, downloading the database, entirely or partly, from the remote host.

The preprocessing is handled by the submodule `pymcfost_process.py` which contains the function `read_dirs_doimages()` that reads the FITS files produced by MCFOST, and some log files containing the parameters of the simulations, all organized in a specific structure of subdirectories. This function thus convolves the images with Gaussian beams of given sizes and adds some of the read parameters to the header file to reach the complete list in Table 5.1. Therefore, we used this module to preprocess and standardize the previously computed simulations collected, before adding them to the final database. In order to simplify the use of this tool, I wrote a script called `pymc_prepro` which allows the use of the functions just mentioned through a command line interface (CLI).

The second function of these scripts is the creation and handling of the index. The two submodules `index.py` and `index_keys.py` fulfill this purpose. They provide functions that read the parameters stored in the FITS files and generate or update the index. Furthermore, they allow to access the data stored in these index files, `index.js` and `data.js`, through the function `get_data()` which merges their content together, returning a python dictionary. This dictionary can then be passed to a function of `pandas`, a python library designed for data analysis and manipulation, obtaining a more versatile pandas `DataFrame`. The generation of the index files can be done through the CLI script `generate_index` which has to be run from the terminal specifying the directory where the unindexed FITS images are stored.

Last, the third function of these scripts, access and retrieving the dataset from the remote server where it is hosted, is carried out by the `downloader` module. It implements functions which are able to determine the remote location of specific items of the database given their unique keys, download them and check whether the download has succeeded. The code developed employs the `https` protocol to get the data and compute the `sha1` checksum to check the integrity of the files retrieved. These functions are interfaced by the CLI script `get_db`.

5.4 Expanding the dataset

The version of the dataset deployed with this thesis is not intended to be the final version. We have seen, in the previous section, that only a small set of all the parameters exhibit different values distributed across a sufficiently large range. Variations of the other ones could be explored in future updates.

In the development of the database structure, great effort has been put into making the addition of new data as easy as possible. The previously discussed scripts were also developed with this purpose.

In order to integrate new images, FITS files obtained from a radiative transfer simulation done with MCFOST or a similar tool are needed. If results of hydrodynamical simulations are available, they also can be used, additionally running the radiative transfer simulations. Before using the `pymc_prepro` script to preprocess the images and convolve them with the Gaussian beam, they need to be organized in a specific tree of directories. Each group of images with the same values for these parameters

key	description
RIN	inner radius of the hydrodynamical simulation domain
ROUT	outer radius of the hydrodynamical simulation domain
RC	reference radius for the exponential taper in the radial profile density
HRIN	aspect ratio at RIN
RHOG	intrinsic grain density
PINDEX	power law index for the initial surface density profiles
QINDEX	power law index for the initial sound speed profile
ALPHASS	Shakura-Sunyaev α-viscosity
ECC	eccentricity of the planet's orbit

have to be grouped inside the same directory organized according to the following substructure:

```
parent_dir
├── MP1_time050
│   ├── gd50
│   │   ├── data_th
│   │   ├── data_1300
│   │   │   ├── RT.fits.gz
│   │   │   └── disc.para
│   │   └── image.log
│   ├── gd70
│   ├── gd100
│   └── ...
├── 2HoR_time100
│   └── ...
├── MP25_time150
│   └── ...
└── ...
```

The different simulation are thus divided in subdirectories according to the planet's mass simulated, the time of the snapshot and the gas to dust ratio. Additionally, inside each of the `gd#` directories there might be different subdirectories named `data_#` automatically created by MCFOST and containing results at different wavelengths (indicated by the # in μm). Apart from the `RT.fits.gz` file produced by the MCFOST there are two other files needed as it can be seen from the previous scheme: `image.log` contains the redirected standard output of the `MCFOST` run which provides the parameters read from the phantom file, `disc.para` is instead the input file used to run that simulation. When all the data available are properly grouped in this way, the `pymc_prepro` script can be run. If some of the features in the last table vary significatively across multiple simulations this script results inconvenient. In this case the functions implemented in the `dblib` package can be used to build a new script adapted to this situation.

Once the FITS images have been preprocessed they can be added to the local directory containing the dataset and indexed running the script `generate_index` with the option `--update`.

At this point the database has been expanded. To share the contribution it is possible to merge it to the public remote database opening a pull request in the hosting repository.

The current version of this database can be improved under many aspects. The first suggestion is, as I anticipated, the addition of simulations exploring ranges of the parameters which do not vary across the images already present. For example, some hydrodynamical variables such as the α-viscosity or the density radial profile could generate different pictures which could then be interesting to probe with a neural network. Furthermore, we only included images of discs where the embedded planet interacted with the medium inducing the formation of gaps. It might be interesting to include in the dataset also simulations where the gap does not form, both with and

without a planet, or, on the contrary, where we observe gaps without the presence of planets. In this way, these images might be used, for example, to train a neural network to detect planets and investigate the existence of ways to confidently distinguish planet induced gaps from substructures due to different phenomena.

Images of discs with double stars or more than one planet might also be included, together with examples of the other possible substructures in order to include all the potential morphologies. With these additions it may also be necessary to update the headers of these files including some flags which would allow a more selective filtering of the data depending on the needs.

Chapter 6

Proof of concept

In order to prove the feasibility of our approach, we used the collected data to develop and train a neural network aimed at inferring planets' masses from the sole flux density map of the dust continuum emission. The model was implemented using TensorFlow's libraries.

In this chapter I am going to present the trained model, providing a usage example of the dataset developed. In this context I will show its versatility in the selection of subsets of data, and the simple steps required to interface it with standard TensorFlow's neural network models. The obtained results will be reviewed discussing the successfull achievements and the emerged limits. Finally, I will try to reach the final purpose of this approach applying the trained model to observations of real discs.

6.1 Data pre-processing

The entire dataset developed contains 24720 images for a total size of approximately 190 GiB. In order to run the training process in reasonable times, remaining able to explore different configurations, we had to select a subset of the entire collection of data. In addition to the overcome of computational limits, this selection is also useful to reduce the redundancy of the entire dataset which, in its completeness, cotains several images of the same disc convolved with different beam sizes that simulate different resolutions of the observative tools. These images were in fact, by design, intended to be filtered according to the expected resolution of the real images object of study.

In this specific case I selected the images convolved with a beam dimension of 0.046 arcsec which is, among the available sizes, the closest to the currently achievable best resolution according to the DHSARP project. To further reduce the size of the dataset, I also selected only images with inclinations of about 60° which constitutes the larger set. As a result, I obtained a set of 1024 images which was randomly split in two subsets for the training and test phases containing respectevely the 80% and 20% of the whole data. The test set contains images which are not used during the training process allowing the evaluation of the model on never seen.

With the aim of showing how the database developed can be used in machine learning applications, I am going through the steps required to filter the data reporting the lines of code used.

First, the dataset index has to be imported and loaded using the function `dblib.index.get_data()` developed for this purpose, which returns a python dictionaries. It that can then be loaded in a pandas `DataFrame` which allows a simpler filtering using pandas queries:

```
1    import pandas as pd
2    import dblib.index as index
```

```
3
4       data = pd.DataFrame(index.get_data())
5       data = data.set_index('KEY')
6       trial_data = data[(data.BMAJ>0.046) & (data.BMAJ<0.048)
7                           & (data.INCL>58)][['FILENAME', 'MPLANET0']]
```

In these lines, the selection previously discussed is applied to the entire dataset and, for each item, only the name of the file that stores the disc image and the mass of the embedded planet are kept, as all the other parameters will not be used. However, if after the training process, for any reason, becomes necessary to access the other variables they could be retrieved using the KEY parameter which, in line 5, is set as the index column.

The following step requires loading in memory all the images. This operation is achieved through the following instructions

```
1       from astropy.io import fits
2       import numpy as np
3
4       def get_image(filepath):
5           with fits.open(filepath) as hdu:
6               return hdu[0].data
7
8       train_images = np.array(
9                       [get_image(f"../../database/{filename}")
10                      for filename in trial_data['FILENAME'].tolist()]
11                      )
```

where the defined function `get_image()` opens the FITS file and extracts the data using the `io.fits` module of the `astropy` package. All the images are thus stored in a `numpy` array which results three-dimensional with shape `(n, 1024, 1024)` where `n` is the total number of images loaded.

The last step, before the split in training and test sets, consists in the reshaping of this array. The function `get_image()` return a 1024x1024 matrix of float containing the value of the light intensity at each pixel. Tensorflow models require each 2D image to be, instead, a matrix of arrays. The additional dimension of the obtained tensor is introduced to account for different channnels allowing, for example, the analysis of RGB images where the intensity, at each pixel, of the three colors Red, Green and Blue is given.

The data of the images stored in our database has only one channel. Therefore, the array can be reshaped through

```
1       train_images = train_images.reshape(train_images.shape[0],
2       train_images.shape[1], train_images.shape[2],  1)
```

obtaining the new array `train_images` with the shape `(n, 1024, 1024, 1)` where `n` is the total number of items in the set. This array can be directly passed to the `TensorFlow` function that initiates the training process.

These are all the steps required to retrieve the database we developed, filter it according to the application and interface the data with standard and popular python packages.

6.2 Adopted model

This first machine learning application was primarly designed with the purpose of probing the limits and potentials of this new approach. For this reason, we chose the simplest model with the ability to solve the problem addressed among the sundry alternatives availables.

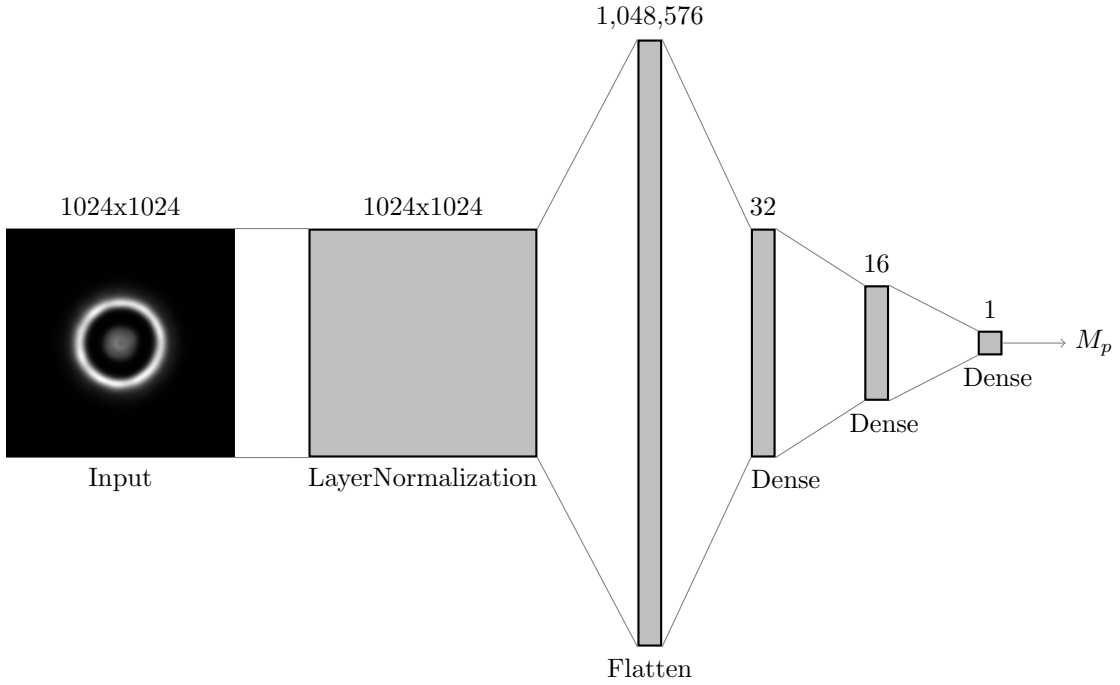

Figure 6.1: Scheme of the neural network model implemented for the proof of concept. For each layer its dimension and type are indicated respectively above and below its representation. This model has a total of 33,557,057 trainable parameters.

More specifically we implemented a feedforward, densely connected, neural network with two hidden layers made of respectively 32 and 16 neurons and an output layer with a single neuron that returns the predicted value of the planet's mass. The ReLU activation function was used in each unit, except for the output one, to introduce non-linearity.

The input interface is made of two layers with specific functions. The input images, containing data organized in two dimensions, are provided to the first layer which is able to standardize the data subtracting the mean value and dividing for the standard deviation. Tensorflow provides two different layers which implement this process with specific differences. The first one is called `LayerNormalization` and computes the mean and standard deviation values indipendently for the flux density data of each image which are thus separately standardized. The alternative is instead called `BatchNormalization`. During the training process, it works using the mean and standard deviation computed with the data of all the images in the current batch of inputs. Batches are groups in which the items in the dataset are divided during the training process, their size can be modified as a hyperparameter. After the training process, when the model is applied to the validation set or to new data, this layer normalizes the input using a moving average of the means and standard deviations of the batches it has seen during training. The first option basically focus on the flux density variations inside each image enlightening the presence of substructures, such as gaps, but, on the other end, removing the information carried by the absolute values of the flux density which are essentially measures of the dust mass. This information, more precisely the relative difference of the dust mass among different discs is instead preserved with the second option which, however, has the risk of reducing the fluctuations inside each image potentially hiding the existence of substructures. Following these arguments, we expected the first option to produce better results and some tests confirmed this hypothesis. Therefore, we used the `LayerNormalization` layer in the implementation whose results are discussed in this chapter.

After the normalisation, the data, organized in two-dimensional matrices are flattened by a layer

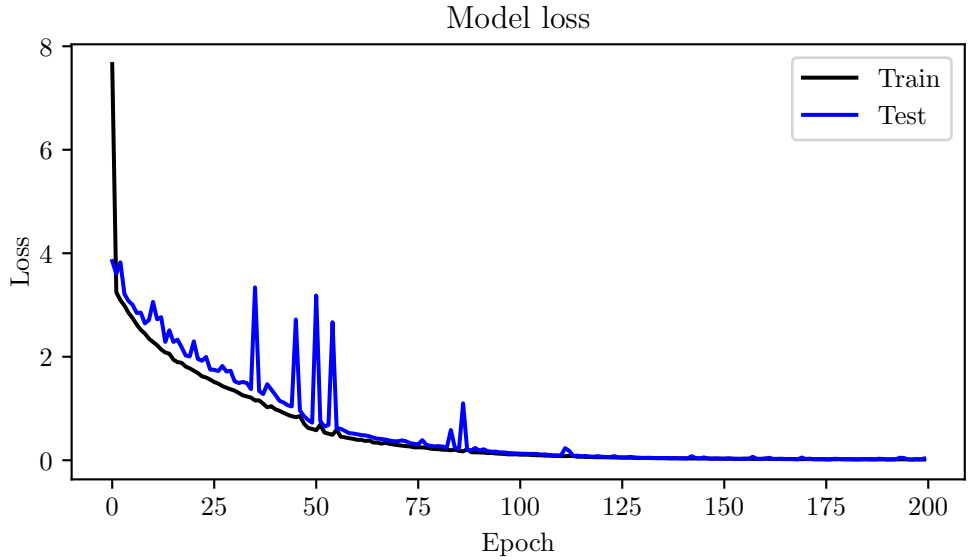

Figure 6.2: Trend of the loss function (mean squared error) evaluated on the training set (black line) and the test set (red line) after each epoch. Both the curves converge towards 0 indicating that the neural network has been successfully trained to predict the planet's masses from images showing discs' morphologies.

aimed at this purpose, before eventually, reaching the hidden layers.

The optimizer used is `Nadam` which is analogous to the `adam` optimizer but implemented with a Nesterov momentum. The `adam` optimizer is a stochastic gradient descent method with the addition of first order and second order moments that are adaptively estimated during the training process. According to Kingma et al., 2014, the method is "computationally efficient, has little memory requirement, invariant to diagonal rescaling of gradients, and is well suited for problems that are large in terms of data/parameter". The Nesterov momentum algorithms differ from the traditional ones essentially for where, at each step, the gradient is evaluated. Using a Nesterov momentum its contribution to the motion in the parameter space is considered before computing the gradient which is evaluated in the new point reached instead of the original one. This simple modification usually boosts the training process allowing a faster convergence. We also lowered the learning rate reaching the value 10^{-4} to mitigate some undesired effects encountered.

All these hyperparameters, including the number of hidden layers and of their neurons, were accurately chosen with a trial and error approach. Here, we presented the model which showed the best performance preserving a relative simplicity and thus economic computational requirements. Figure () shows an overview of its main features.

To evaluate the model, both during and after the training process, we used, as the loss function, the mean squared error of the predicted mass values with repect to the ones known from the simulation files and stored in our dataset.

6.3 Results

The training of the previously presented model took about 5 hours of wall time running on a laptop equipped with a dual core CPU with 2GHz clock frequency and an NVIDIA GeForce 840M GPU. Additionally considering the opportunity to run this process on a faster and better performing machine, the computational time required by the training of a neural network model appears to be significantly smaller than that needed for numerical simulations. We thus proved the first element

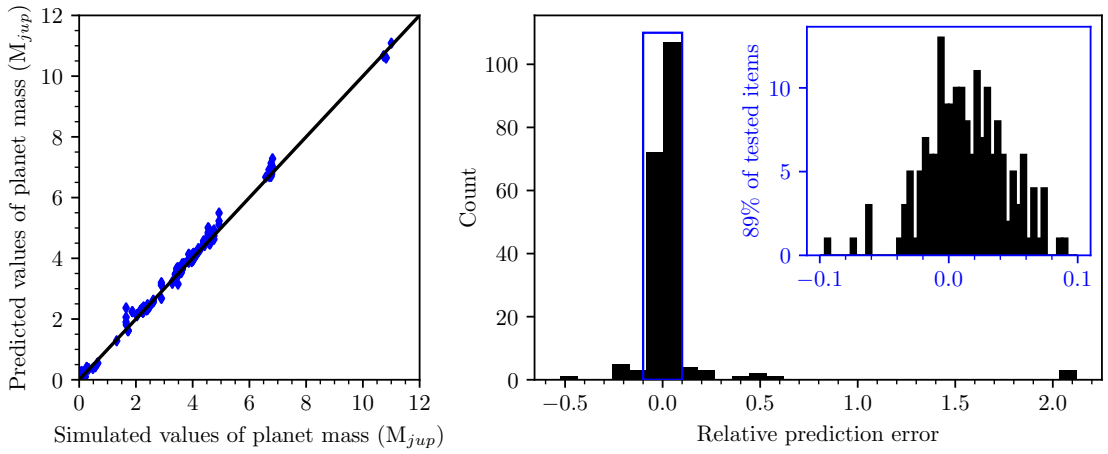

Figure 6.3: Left panel: correlation between the predicted mass and the real value obtained from the simulation files for all the items in the test set. The black lines draws the $M_{pred} = M_{sim}$ curve. Right panel: Distribution of the relative prediction error $(M_{pred} - M_{sim})/M_{sim}$ computed for the test set. The red box zooms in the -10% to 10% range which contains the 89% of all the items.

in support of machine learning approaches.

The neural network was trained for 200 epochs. The trends of the loss function evaluated on the training and test sets after each epoch are shown in figure () where their drop, after approximately 125 epochs can be appreciated. Furthermore, there are not evident signs of underlearning or overlearning as the loss function keeps low values for both the samples. More precisely, low values of the mean squared error computed predicting the masses of planets in the test set images show the successfull functioning of the trained model on new unseen data.

This graph can, alone, establish the fullfilment of this first application purpose which was to evaluate the feasibility of a machine learning approach. However, we further analyzed the obtained results to provide an estimation of the model's accuracy also comparing it with the results obtained by Auddy and Lin with their approach.

The root mean squared error evaluated on the test set could be assumed as the prediction uncertainty of the trained model which is $0.15 M_{Jup}$ corresponding to respectively the 15% and 1.5% of the less and the most massive planets for which it has been tested. Auddy and Lin achieved with their model an uncertainty estimated, with the same method, in the 11% of the most massive planet on which they tested their network. Due to the different ranges of planet's masses adopted the comparison is possible only through the relative error.

The left panel of figure () shows the correlation between the predicted and simulated masses of the validation set which lie along the $M_{predicted} = M_{simulated}$ line indicating a strong correlation. We also computed the relative prediction error $(M_{pred} - M_{sim})/M_{sim}$ for each item in the test set. The right panel of the same figure illustrates its distribution presenting a mean value and standard deviation respectively of 5% and 27%. The trained model is afflicted by a slight bias which tends to overestimate the planet masses. Furthermore, the relative error distribution presents a large standard deviation which indicates the great variability of this quantity in its evaluation among the test set. For instance there are a few cases afflicted by prediction errors up to 200%.

There are hence some margins for improvements partly due to the reduced size of the dataset and the low complexity of the neural network employed. With more sophisticated models and a larger dataset we are confident that better results could be obtained.

Finally, we also tried to apply the trained model in a real-case scenario. In this last test we used

the ALMA observation of the DS Tau disc in the dust continuum at 1.3 mm. The image, obtained processing the raw data, was provided by () We further adapted it to meet the same specifics of the ones collected in the employed dataset. First it was cropped to approximately the same size considering the pixel scale for each axis. We then checked that the data was in the same units and upscaled the image using `scikit-image` functions to have the same resolution of the images in the training set (1024x1024). The obtained matrix of data was thus passed to the trained model which predicted a planet's mass of $M_p = 3.3 M_{Jup}$. Using numerical simulations () established the presence of a planet, hidden in the gap, with an estimated mass of $3.5 \pm 1 M_{Jup}$. The two results are thus compatible.

Chapter 7

Conclusions

7.1 Conclusion

7.2 Future perspectives

-server side -noise through nn

Acknowledgements

Bibliography

[1] Philip J. Armitage. *Astrophysics of planet formation*. Cambridge university press, 2010.

[2] Benedetta Veronesi, Enrico Ragusa, Giuseppe Lodato, Hossam Aly, Christophe Pinte, Daniel J Price, Feng Long, Gregory J Herczeg, and Valentin Christiaens. Is the gap in the DS Tau disc hiding a planet? *Monthly Notices of the Royal Astronomical Society*, 495(2):1913–1926, 05 2020.

10.1093/mnras/stz913

nadam citation: http://arxiv.org/abs/1412.6980