



UNIVERSITÀ DEGLI STUDI DI MILANO
FACOLTÀ DI SCIENZE E TECNOLOGIE

Corso di Laurea in Fisica

**Cosmological applications of
symbolic regression algorithms**

Relatore: **Prof. Luigi Guzzo**

Relatore Esterno: **Dott. Marco Bonici**

Tesi di Laurea Triennale di:
Francesco Farinelli
Matricola: 966796

Anno Accademico 2022/2023

A Sofi
2007-2023 †

Contents

Introduction	III
1 Cosmology	1
1.1 General relativity	1
1.2 Dark Matter	5
1.3 Λ CDM model	7
1.4 The Cosmic Microwave Background radiation	8
1.5 Structure formation	9
1.6 Power spectra	12
1.6.1 Matter power spectrum	12
1.6.2 CMB power spectrum	14
1.7 Matter transfer function	15
1.8 Einsten-Boltzmann equations	15
2 Machine learning	17
2.1 Neural networks	18
2.2 Symbolic Regression	23
2.2.1 Genetic algorithms	24
2.3 PySR	25
2.3.1 Tests on PySR limits	26
3 Application of symbolic regression to $T(k)$	30
3.1 Dependence of $P(k)$ on cosmological parameters	30
3.1.1 Baryonic matter	31
3.1.2 Cold dark matter	31
3.1.3 Massive neutrinos	32
3.2 Data	33
3.2.1 Baryons and CDM	33
3.2.2 Baryons, CDM and massive neutrinos	34
3.3 Performance evaluation metrics	34
3.4 Fitting formulae from PySR	34
3.4.1 Baryons and CDM	35
3.4.2 Baryons, CDM and massive neutrinos	38
3.5 Comparison with BBKS and EH formulae	39
3.6 Validation test	40

4 Application of symbolic regression to C_ℓ^{EE}	42
4.1 Dependence of CMB power spectrum on τ_{reio}	42
4.2 Data	44
4.3 Fitting formulae from PySR	44
4.3.1 SR on CLASS data	46
4.3.2 SR on preprocessed CLASS data	47
4.4 Determination of a general formula for the ratios	49
5 Conclusions	51
A BBKS formula	54
B EH formulae	55
B.1 No massive neutrinos	55
B.2 Considering massive neutrinos	56
Bibliography	58

Introduction

The Universe we live in exhibits deviations from perfect homogeneity and isotropy. These deviations manifest in the form of various celestial objects, such as galaxies and clusters of galaxies, which clump together due to gravitational interactions.

The evolution of the Universe finds a well-established description in the Λ CDM model, which is able to fit different astronomical observations, including the large-scale distribution of galaxies and the Cosmic Microwave Background (CMB). In order to analyze these data, cosmologists compute quantities such as the matter power spectrum $P(k, z)$, which characterizes the matter density fluctuations of the universe, and the CMB angular power spectra C_ℓ , which characterize the angular fluctuations in CMB measurements.

The scale dependence of the matter power spectrum is encoded in the transfer function $T(k)$, which models the evolution of the primordial density fluctuations and is numerically computed using programs known as Einstein-Boltzmann solvers, due to the absence of an analytical solution to the governing differential equations for $T(k)$. In a similar fashion, also the CMB angular power spectra are numerically computed by the same kind of programs.

In recent years, Machine Learning (ML) has gained significant importance due to its ability to effectively address scientific problems. ML, a subset of Artificial Intelligence (AI), is dedicated to designing algorithms and models that strengthen computers to acquire knowledge and make predictions or decisions based on data. Rather than relying on explicit programming, machines derive insights from data patterns. In order to accelerate their analyses, the astrophysical community has developed emulators, codes based on ML tools such as Neural Networks (NN), to accurately describe the dependence of quantities such as $T(k)$ and C_ℓ on cosmological parameters. Although the success of this class of tools, given their precision and computational performance, they do have some shortcomings such as the lack of interpretability of the trained models.

An alternative and possibly complementary route to address the same problem is represented by Symbolic Regression (SR). SR explores the space of mathematical expressions to identify the most suitable fit for a given dataset. SR involves a supervised learning task where the model space is defined by analytic expressions. The primary advantage of SR over traditional NN lies in its capacity to provide an interpretable model, as opposed to the black-box output characteristic of conventional ML techniques like NN.

In this thesis we will try to apply SR to the two problems described above, to see if SR could be a good tool in this field of physics.

Chapter 1

Cosmology

1.1 General relativity

According to General Relativity (GR), the equations that describe our universe in natural coordinates are the Einstein field equations:

$$G_{\mu\nu} = 8\pi T_{\mu\nu}. \quad (1.1)$$

Those equations relate the Einstein tensor $G_{\mu\nu}$ and stress-energy tensor $T_{\mu\nu}$, that is a tensor physical quantity that describes the density and flux of energy and momentum in spacetime [1]. Einstein tensor itself is defined by:

$$G_{\mu\nu} := R_{\mu\nu} - \frac{1}{2}Rg_{\mu\nu}, \quad (1.2)$$

where $R_{\mu\nu}$ is the Ricci tensor and R is the Ricci scalar: they are two geometrical objects defined by the Riemann curvature tensor $R^\sigma_{\rho\mu\nu}$. This is the most common way used to express the curvature of Riemannian manifolds (our universe in this case); so, equations 1.1 relate the geometry of spacetime to the distribution of matter within it.

Which solution of Einstein's equations describes our universe? To answer this question, we need some hypotheses motivated by observations. We will make the hypothesis that the universe is homogeneous and isotropic: this is the Cosmological Principle. It is a reasonable assumption that our position in the universe is not privileged. If we were in another region of the universe, the basic characteristics of our surroundings would be the same, *i.e.* the universe is homogeneous. It is also convenient to assume that the universe is isotropic, that is, such that no spatial direction is preferred.

In general, we can have a universe with a positive curvature ($k > 0$), a negative curvature ($k < 0$) or a flat universe ($k = 0$). In 1949, Luther Eisenhart [2] showed that the spatial geometry of this kind of universe, described by 1.1, can only be hyperbolic ($k = -1$), spherical ($k = 1$) or flat ($k = 0$). This implies a metric of this form:

$$ds^2 = -d\tau^2 + a^2(\tau)[d\psi^2 + f_k^2(\psi)(d\theta^2 + \sin^2\theta d\phi^2)] \quad (1.3)$$

$$f_k(\psi) = \begin{cases} \psi & k = 0 \\ \sin \psi & k = 1 \\ \sinh \psi & k = -1 \end{cases}, \quad (1.4)$$

where $a(\tau)$ is the scale factor, determined by Einstein equations: 1.3 is called Friedmann–Lemaître–Robertson–Walker metric (FLRW): to get $a(\tau)$ we have to substitute 1.3 into Einstein equations: we obtain this system of equation:

$$\begin{cases} \frac{3\dot{a}^2}{a^2} = 8\pi\rho - \frac{3k}{a^2} \\ \frac{3\ddot{a}}{a} = -4\pi(\rho + 3p) \end{cases}. \quad (1.5)$$

These are called Friedmann equations, where p is the pressure and ρ is mass density of matter. We can see that if $p \geq 0$ and $\rho > 0$, universe can't be static: it must be in contraction or expansion.

The Hubble law

Hubble's law is considered the first observational basis for the expansion of the universe. In fact, he showed that galaxies are moving away from Earth at speeds proportional to their distance. Let us consider r as the distance between Earth and a certain galaxy: we know that the distance is given by the metric (assuming $d\tau = 0$ on a timelike surface):

$$r(\tau) = a(\tau) \int d\Omega, \quad (1.6)$$

where $d\Omega^2 = d\psi^2 + f_k^2(\psi)(d\theta^2 + \sin^2 \theta d\phi^2)$, and

$$r(\tau + d\tau) = a(\tau + d\tau) \int d\Omega = a(\tau + d\tau) \frac{r(\tau)}{a(\tau)}. \quad (1.7)$$

With a Taylor expansion of 1.7 we obtain:

$$v = \frac{dr}{d\tau} = \frac{\dot{a}}{a} r := Hr \quad (1.8)$$

where $H := \frac{\dot{a}}{a}$ is the Hubble parameter. Hubble's law has been confirmed from the observation of the redshift of distant galaxies, for which we have that $\dot{a} > 0$. It is convenient to define the reduced Hubble constant $h := \frac{H_0}{100}$, where H_0 is the present day Hubble parameter, namely the Hubble constant, $H_0 = H(\tau = \tau_0) = \frac{\dot{a}}{a}|_{a=1}$.

Expansion of the universe

Most of the scientific community agrees that the universe began with the Big Bang, a theory that describes how the universe expanded from an initial state of high density, curvature and temperature. The early phase of life of the universe is often referred as Inflation, a period in which the universe was permeated by a very high energy density, and was rapidly expanding and cooling.

After Inflation, and until about 47.000 years after the Big Bang, the dynamics of the universe were set by radiation, *i.e.* everything that moves at relativistic speeds, like neutrinos or photons. Assuming a description of the universe by the FLRW metric 1.3, we obtain that for a radiation dominated universe

$$a(\tau) \propto \tau^{\frac{1}{2}}.$$

Between about 47.000 years and 9.8 billion years after the Big Bang, the energy density of matter exceeded both the energy density of radiation and the dark energy density. In this context we have

$$a(\tau) \propto \tau^{\frac{2}{3}}.$$

The last step, from 9.8 billion years after the Big Bang to present day, is often referred as dark energy dominated era. Although we can think that the gravity would force the universe to remain at least restrained, recent measurements show that the universe is expanding: this is thought to be caused by the negative pressure associated with the cosmological constant, that increasingly dominates over opposing gravity. So, now we have

$$a(\tau) \propto \exp(H_0\tau).$$

This exponential dependence on time makes the space-time geometry identical to the de Sitter universe [3], and only holds for a positive sign of the cosmological constant.

Critical density

The three entities that we will need to describe energy density of the universe are matter, with $\rho_m \sim a^{-3}$, radiation, with $\rho_r \sim a^{-4}$ and cosmological constant, with $\rho_\Lambda \sim \text{const}$. In a flat universe the total energy density today must sum to match the Hubble constant, as we can see inverting equations 1.5. This is referred to as the critical energy density:

$$\rho_{crit,0} = \frac{3}{8\pi} H_0^2 \quad (1.9)$$

where the 0 stands for present epoch. We use this to define dimensionless density parameters for each fluid component,

$$\Omega_{w,0} = \frac{\rho_{w,0}}{\rho_{crit,0}}, \quad (1.10)$$

So it is clear that in general the dimensionless parameters sum to

$$\sum_{w=m,r,\Lambda} \Omega_{w,0} = 1 + \frac{k}{a^2 H_0^2}. \quad (1.11)$$

In particular, if we are in a flat universe then we must have $\sum_w \Omega_{w,0} = 1$. Any excess energy density, with $\sum_w \Omega_{w,0} > 1$ means that we necessarily live in a positively curved universe with $k = +1$. Any deficit in the energy, with $\sum_w \Omega_{w,0} <$

1 gives rise to a negatively curved, $k = -1$ universe. Sometimes it is useful to define density parameter of curvature as

$$\Omega_{k,0} = -\frac{k}{a^2 H_0^2}, \quad (1.12)$$

to obtain the relation

$$\sum_{w=m,r,\Lambda,k} \Omega_{w,0} = 1. \quad (1.13)$$

Cosmological redshift

The cosmological redshift is the shift of spectral lines to longer wavelengths associated with the isotropic expansion of the universe. The redshift z is defined to be

$$z = \frac{\lambda_o - \lambda_e}{\lambda_e}, \quad (1.14)$$

where λ_e is the wavelength of the line as emitted and λ_o the observed one. The redshift has a deep meaning in cosmology [4], which we can demonstrate looking at the expression 1.3 for the metric. In fact, the frequency of a light signal that comes from a point p_1 in the spacetime with k^μ as wave vector measured from an observer with 4-velocity u^μ at a point p_2 is

$$\omega = -k_\mu u^\mu. \quad (1.15)$$

We need to define a special category of vector fields: a Killing vector field, named after Wilhelm Killing, is a vector field on a Riemannian manifold that preserves the metric. Killing fields are the infinitesimal generators of isometries. It can be shown that we can find a Killing vector ξ^μ which allows us to write

$$k_\mu = \alpha \left(u_\mu + \frac{\xi_\mu}{\sqrt{\xi^2}} \right), \quad (1.16)$$

but k^μ is a light-like vector, so $k^\mu k_\mu = 0$, which implies

$$\left(u_\mu k^\mu + \frac{\xi_\mu k^\mu}{\sqrt{\xi^2}} \right) = 0 \implies -u_\mu k^\mu = \frac{\xi_\mu k^\mu}{\sqrt{\xi^2}} \implies \omega_{1,2} = \frac{\xi_\mu k^\mu}{\sqrt{\xi^2}} \Big|_{p_{1,2}}. \quad (1.17)$$

Recalling that $\xi_\mu k^\mu \Big|_{p_1} = \xi_\mu k^\mu \Big|_{p_2}$ because ξ is a Killing vector and k is tangent to the geodesic [1] and $\sqrt{\frac{\langle \xi, \xi \rangle \Big|_{p_2}}{\langle \xi, \xi \rangle \Big|_{p_1}}} = \frac{a(\tau_2)}{a(\tau_1)}$, we obtain

$$\omega_{1,2} = \frac{\xi_\mu k^\mu}{\sqrt{\xi^2}} \Big|_{p_{1,2}} \implies \frac{\omega_2}{\omega_1} = \frac{a(\tau_1)}{a(\tau_2)}. \quad (1.18)$$

Since $a(\tau_2) = 1$ [4], we finally obtain that

$$z = \frac{\lambda_2 - \lambda_1}{\lambda_1} = \frac{\omega_1}{\omega_2} - 1 = \frac{a(\tau_2)}{a(\tau_1)} - 1 \implies a(\tau_1) = \frac{1}{1+z}. \quad (1.19)$$

So, redshift is a measure of the scale factor of the universe when the radiation was emitted by the source. When we observe a galaxy with redshift $z = 1$, the scale factor of the Universe when the light was emitted was $a(\tau) = 0.5$, that is, the distances between fundamental observers (or galaxies) were half their present values.

The Cosmological constant

Einstein's cosmological constant Λ is the constant coefficient of a term that Albert Einstein added to his field equations of GR, because 1.1 have no static solution, which was the one sought by Einstein. So in this case, 1.1 become

$$G_{\mu\nu} = 8\pi T_{\mu\nu} + \Lambda g_{\mu\nu}. \quad (1.20)$$

Even if these are more complete equations, the fact that the universe is expanding, proved by Hubble, ruled out static solutions, pushing Einstein to discard Λ . The discovery that the expansion of the universe is accelerating implies that Λ has a positive value: assuming the cosmological principle, around 74% of the mass-energy density of the universe can be attributed to the cosmological constant Λ , the simplest possible explanation for the so-called Dark Energy (DE). In this situation, the first Friedmann equation becomes

$$\frac{3\dot{a}^2}{a^2} = \frac{8\pi}{3}\rho - \frac{\Lambda}{3} - \frac{3k}{a^2}. \quad (1.21)$$

This equation describes the expansion of the universe as a function of the curvature parameter k , cosmological constant Λ and mass density of matter ρ . If we look at the solutions, we could observe that the curvature has a crucial impact: in fact if the geometry is spherical, *i.e.* the curvature is positive, the expansion is destined to decelerate: this fact will lead in the future to the collapse of the entire Universe into a singularity, the Big Crunch. Universes with hyperbolic or flat geometry, on the other hand, expand respectively at an accelerated rate or with a constant rate. The term containing the cosmological constant Λ tends to make all solutions expansive and is the term that currently seems to dominate.

1.2 Dark Matter

Dark Matter (DM) is a form of matter thought to account for approximately 85% of the matter in the universe. By definition, DM is anything that does not interact through electromagnetic force. It means that DM neither emits nor absorbs light.

The need to include this enigmatic concept arises from its central role in correcting the inadequacies of the prevailing standard model, which would otherwise fail to accurately describe observed phenomena in the surrounding cosmos. This contention is supported by compelling empirical evidence, most notably the discrepancies in the rotation curves of galaxies.

A Keplerian decrease in the rotation curves for galaxies is expected; however, for many galaxies a strong incompatibility between theoretical and experimental

data is observed, as we can see in figure 1.1. Experimentally it is observed that v remains constant up to radii much greater than the radius of the galaxy. If we consider a model that includes the notion of DM, we can demonstrate that for large R to have a constant velocity the mass of DM must grow linearly with the radius. There are various hypotheses about what DM could consist of, which can essentially be grouped into two categories, baryonic and non-baryonic DM.

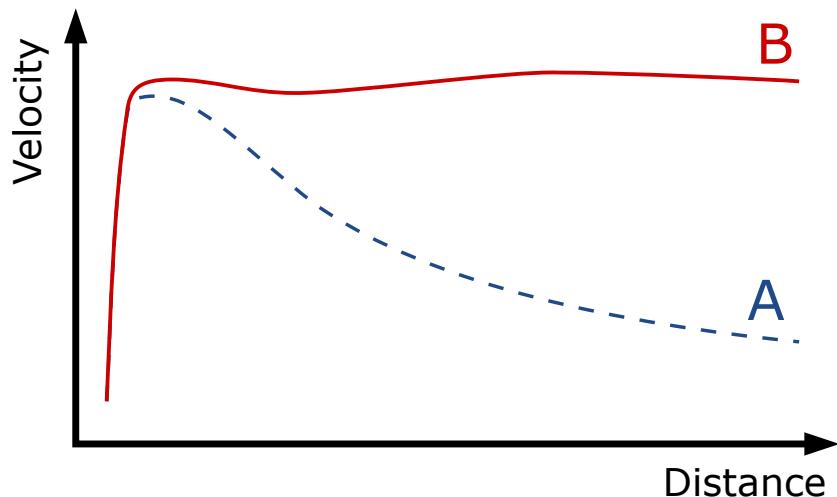


Figure 1.1: Schematic cartoon representation of a typical rotation curve of a spiral galaxy: predicted (A) and observed (B). DM can explain the 'flat' appearance of the velocity curve out to a large radius. Credits: [5]

Baryons are the most obvious candidate for DM, as baryonic DM is observed in the form of planets, low luminosity stars, and diffuse gas clouds. Thus, one may conjecture that DM could be in some form of collapsed objects, that have become known as MAssive Compact Halo Objects (MACHOs). However, direct observations of X-rays from interacting galaxies do not show evidence for such massive halos, indicating that DM does not interact in the same fashion as baryons. Thus, although there is a significant fraction of 'dark baryons' they can not compose more than 15% of the total dark matter, that is dynamically detected. It is therefore unavoidable that the majority of DM be non-baryonic.

Non-baryonic DM can be broadly split into two categories: Hot Dark Matter (HDM), a form of DM which consists of low mass particles that travel with ultra-relativistic velocities, and Cold Dark Matter (CDM) which is supposed to consist of massive and slow moving particles. CDM has long been the leading candidate for what this missing mass in the universe is, but there is no currently no detection of such a particle.

1.3 Λ CDM model

The standard model of cosmology is the so-called Λ CDM model. According to this model, the universe is spatially flat ($k = 0$), contains matter, both ordinary and dark, radiation, and a cosmological constant Λ .

The hot Big Bang scenario assumes that the Universe is expanding. The evidence supporting this is possibly one of the earliest to be confirmed, as we already said, with Hubble's first measurements in the 1920s, and increasingly precise measurements of the rate of expansion in the following century. The Λ CDM model falls into the category of hot Big Bang cosmology and specifies two main aspects: a cosmological constant $\Lambda \neq 0$ and a DM composed of non-relativistic particles from very early in cosmic history that interact only gravitationally. The fact that the DM must be CDM is independent of Λ . Figure 1.2 describes schematically the history of the universe, in the Big Bang scenario with a cosmological constant Λ and CDM.

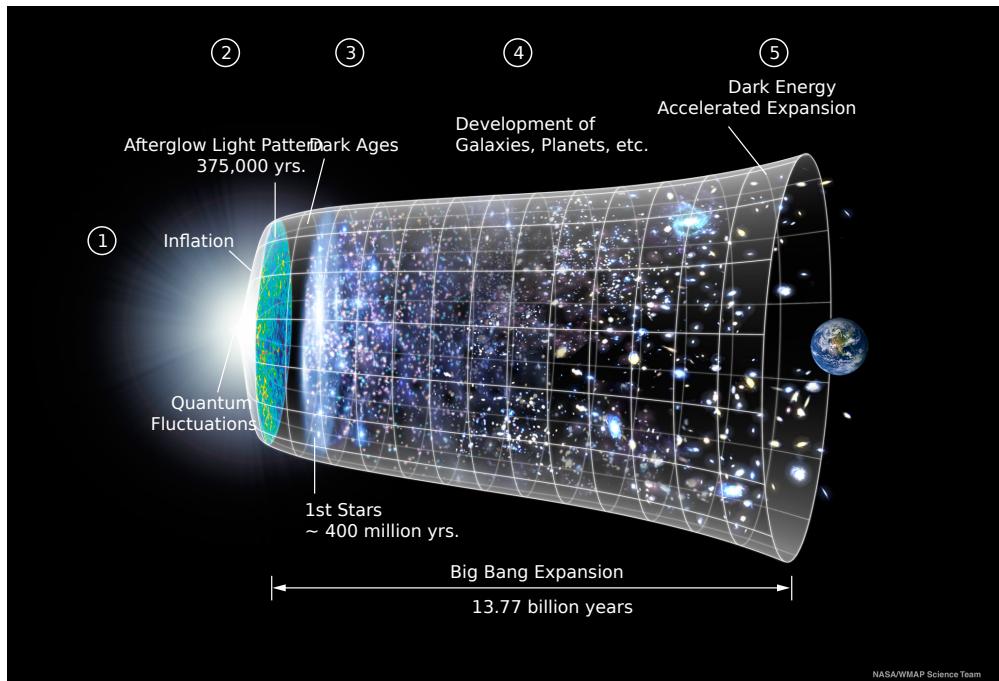


Figure 1.2: In this figure are represented in a brief and simplified form the five theorized stages in the evolution of the universe. Credits: [6]

At the present time sufficiently precise measurements of the cosmological parameters indicate that two components are dominating, the cosmological constant and the matter: $\Omega_\Lambda \simeq 0.69$ and $\Omega_m \simeq 0.31$. The most recent measurements set a bound $|\Omega_k| < 0.01$ on the curvature indicating that we live in a flat universe. This collection of numbers, Ω_w , defines the Λ CDM model. It is convenient to define the reduced density parameters: they are used to simplify and standardize cosmological analysis so that cosmological systems of different sizes and time scales can be compared:

$$\omega_w := h^2 \Omega_w. \quad (1.22)$$

Table 1.1 gives the values for the principal density parameters, as reported from Planck Collaboration [7].

Parameter	Symbol	Value
Reduced Hubble constant	h	0.6770 ± 0.03
Total matter density	ω_m	0.14240 ± 0.00087
Baryonic matter density	ω_b	0.02242 ± 0.00014
Total radiation density	ω_r	$2.47 \cdot 10^{-5}$
Neutrinos density	ω_ν	< 0.0076 , see 1.23
Cosmological constant density	Ω_Λ	0.6889 ± 0.0056

Table 1.1: Cosmological parameters of Λ CDM model

Of these parameters, only ω_r is accurately measured directly. The COBE FIRAS (Far-Infrared Absolute Spectrophotometer) experiment [8] determined CMB temperature, setting $\omega_r = 2.47 \cdot 10^{-5}$. Spatial curvature does not appear in the list, because it can be derived from the other parameters using 1.11. The density parameter for neutrinos is

$$\omega_\nu = \frac{\sum m_\nu}{94 \text{ eV}}, \quad (1.23)$$

where the sum is over all families with mass in $5 \cdot 10^{-4} \text{ eV} \leq m_\nu \leq 1 \text{ MeV}$ range.

Before moving on, let us notice that although this model is highly successful, there are some potential discrepancies, such as the H_0 and σ_8 tensions [9][10], which are still being debated in the scientific community. This fact should not scare us, in fact it is meaning that we have to continue to study this model and we have to understand how it can be improved.

1.4 The Cosmic Microwave Background radiation

The Cosmic Microwave Background radiation (CMB) represents the lingering afterglow of the universe's earliest light, which was able to traverse the cosmos freely. This microwave radiation results from the universe's expansion and the cosmological redshift that followed its initial emission. This ancient radiation, often likened to a cosmic fossil, was released roughly 380,000 years post the Big Bang. Before then, in the pre-recombination era, baryonic matter and radiation co-existed in the form of a plasma. During this period, atoms could not exist in their bound state due to the energetic collisions with free photons and the thermal agitation of particles within this plasma [11][4]. The expansion of the universe made it cool down, allowing matter and radiation to separate. CMB originated under these conditions.

On the largest cosmic scale, CMB serves as compelling evidence for the isotropy of the universe [4], providing robust support for the hot Big Bang scenario and the Λ CDM model. Observations show that this radiation is remarkably consistent across the entire sky, characterized by a blackbody spectrum covering wavelengths

from 1 centimeter to 1 meter. The measurements from the COBE satellite [8] represent the most precise instance of a black body found in nature, *i.e.* with intensity accurately described by the Planck formula

$$I(\nu) = \frac{2h\nu^3}{c^2} \frac{1}{\exp(\frac{h\nu}{k_b T}) - 1}, \quad (1.24)$$

where in the case of the CMB the best fit is obtained for a temperature

$$T = 2.7260 \pm 0.0006 \text{ K}. \quad (1.25)$$

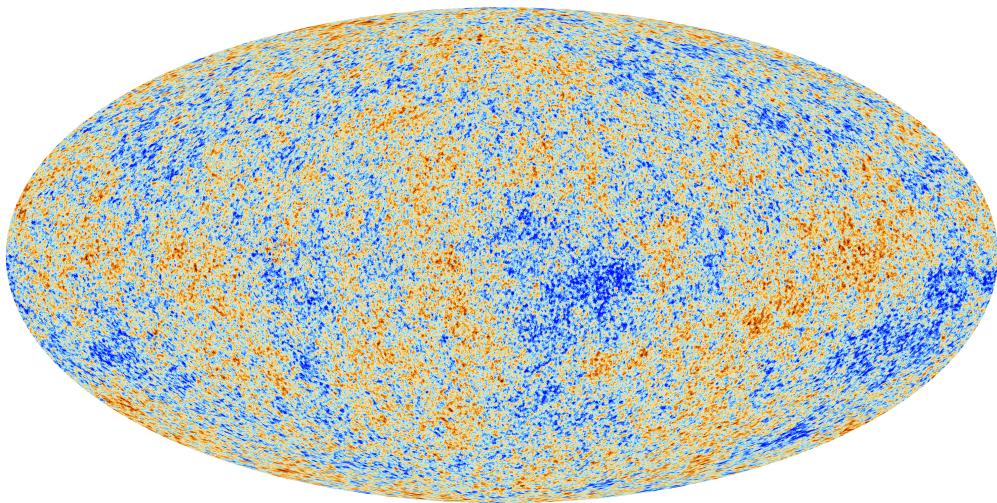


Figure 1.3: The cosmic microwave background is a snapshot of the oldest light in our universe. The colors of the map represent small temperature fluctuations that ultimately resulted in the galaxies we see today. Credits: [12]

As we can see from figure 1.3, there are local deviations from mean value, this at the level of $\frac{\Delta T}{T} \sim 10^{-5}$: these fluctuations are of crucial importance, as they contain key information on the overall seeds that gave origin to the observed structure, which we will investigate in a better way in section 1.5. Without those small irregularities, there would not be any galaxies, and we would not be here to observe them. Likewise, larger anisotropies would not produce the universe we see.

1.5 Structure formation

The Friedmann world models describe a universe that is isotropic and homogeneous [4]: these models do not explain the diverse structures we observe in the universe today, like galaxies and galaxy clusters . To get a better understanding of the actual universe, we need to develop more realistic models. These models should include small variations in matter density within the overall uniformity of the universe. We also need to study how these density variations evolve over time

due to the influence of gravity. This will help us better explain the rich variety of structures we see in the universe.

Now we have to think of universe as a homogeneous and isotropic fluid: in fact, viewed from afar, we should think of galaxies as atoms in a cosmological fluid. We have to introduce the mathematics that stands behind fluid dynamics, and first of all we have to explain the concept of co-moving coordinates: in fact, since the universe is expanding, we need to split the apparent velocity of an object into two contributions, one is its own velocity, and the second is the component of the expansion of the universe, which is not a real velocity, as we saw earlier. So, assuming that \mathbf{r} , $\mathbf{u} = \dot{\mathbf{r}}$ are the physical coordinates and \mathbf{x} , $\mathbf{v} = a\dot{\mathbf{x}}$ are the co-moving coordinates, we have the following relations:

$$\mathbf{r}(t) = a(t)\mathbf{x}(t), \quad (1.26)$$

$$\mathbf{u} = H\mathbf{r} + \mathbf{v}, \quad (1.27)$$

Where H is the Hubble parameter. Considering fluids that live in a static space-time, so neglecting the expansion of the universe, these are described by three equations. The first is the continuity, or Navier-Stokes equation that describes mass conservation,

$$\frac{\partial \rho}{\partial t} + \nabla(\rho \mathbf{v}) = 0, \quad (1.28)$$

the second one is Euler equation, that is Newton's second law $\mathbf{F} = m\mathbf{a}$ for a continuous fluid

$$\frac{\partial \mathbf{v}}{\partial t} = -\frac{1}{\rho} \nabla P - \nabla \Phi, \quad (1.29)$$

and the last one is Poisson equation connecting the gravitational potential $\Phi(\mathbf{r}, t)$ to the density field $\rho(\mathbf{r}, t)$

$$\nabla^2 \Phi = 4\pi \rho. \quad (1.30)$$

Jeans' instability

Let us perturb this homogeneous and isotropic fluid: we will consider adiabatic perturbations, which means that these perturbations are the same for both baryonic and DM. Thus we write

$$\rho(\mathbf{x}, t) = \bar{\rho} + \Delta\rho(\mathbf{x}, t). \quad (1.31)$$

Linearizing equations 1.28 and 1.29 and combining them, we obtain [11] a wave equation with a dispersion relation

$$\omega^2 = c_s^2 k^2 - 4\pi \bar{\rho} = c_s^2(k^2 - k_J^2), \quad (1.32)$$

where ω is the pulsation, k is the wavenumber and c_s is the sound velocity. We have also defined the Jeans wavenumber k_J ,

$$k_J = \sqrt{\frac{4\pi \bar{\rho}}{c_s^2}}. \quad (1.33)$$

This quantity makes us clear that ω^2 can be a positive or a negative number. We know that an $\omega^2 < 0$ figures out a unstable exponential solution, which is very interesting because it determines the rapid growth of the perturbations. An $\omega^2 > 0$ generates an oscillatory solution, as we expect. So, any perturbation that has a size larger than the Jeans length,

$$\lambda_J = \frac{2\pi}{k_J} = \sqrt{\frac{\pi c_s^2}{\bar{\rho}}}, \quad (1.34)$$

will typically grow exponentially quickly due to the effect of gravity. This is known as the Jeans' instability.

The Growth of Perturbations

Now, we want to take account of the expansion of the universe. We report the adapted Navier-Stokes equation,

$$\frac{\partial \rho}{\partial t} + 3H\rho + \frac{1}{a}\nabla(\rho\mathbf{v}) = 0. \quad (1.35)$$

This form makes it clear that if we restrict to solutions in which $\mathbf{v} = 0$, so the velocity of the fluid simply follows the expansion of space-time, then we recover the familiar solution $\rho(t) \sim a^{-3}$, which simply tells us that the density dilutes as the universe expands. We can get a single equation [11] telling us how the density perturbation $\Delta = \frac{\Delta\rho}{\bar{\rho}}$ evolves in an expanding space-time

$$\ddot{\Delta} + 2H\dot{\Delta} - c_s^2\left(\frac{k^2}{a} + k_J\right)\Delta = 0. \quad (1.36)$$

Let us notice that now we are working in Fourier space, and so $\Delta = \Delta(\mathbf{k}, t)$: the advantage of working in this space is that the equation is decomposed into a separate equation for each value of \mathbf{k} . The second term $2H\dot{\Delta}$ modifies the analysis in a crucial way: this is referred to as Hubble drag, which slows the peculiar velocities down as the universe expands.

Solutions to 1.36 have a strong dependence from the Jeans' wavelength λ_J : in fact, note that the equation 1.36 is in co-moving coordinates, $k_{phy} = k_{comov}/a$, and k_J is the physical Jeans wavenumber, so we have two cases. If $\lambda < \lambda_J$ we obtain a damped harmonic oscillator, with Hubble drag that provides the friction term. On the other hand, with $\lambda > \lambda_J$ our solutions will suffer Jeans' instability: but this facts is a good thing, because we obtain solution that grows over time, and in this way we have a mathematical support for structure formation. The expansion of the universe set another length scale,

$$\lambda_H = \frac{c}{H}, \quad (1.37)$$

which is called *apparent horizon*, that identifies the apparent dimension of the universe. Now we can take a quick look at the growth of matter density perturbations in the different dominated eras of our universe.

In a matter dominated universe, and so in a non-relativistic fluid regime, we know that Jeans' length sits well within the horizon, $\lambda_J \ll \lambda_H$, and so we obtain [11] for solutions with Jeans instability

$$\Delta(\mathbf{k}, t) \sim \begin{cases} t^{\frac{2}{3}} \\ t^{-1} \end{cases}. \quad (1.38)$$

We see how the expansion of the universe slows down the growth of the Jeans instability which is not exponential anymore, reducing the rate at which objects undergo gravitational collapse into a power-law.

In a radiation dominated universe we have to change a little bit the point of view, because now our problem is set in a relativistic fluid. The three equation 1.28, 1.29 and 1.30 can be modified accordingly. In this situation we obtain a regime in which $\lambda_J \sim \lambda_H$, and we learn that, during the radiation dominated era, the matter perturbations inside the horizon grow as $\Delta \sim \log t$.

In a cosmological constant dominated universe it can be shown that there is no growth of perturbations. Solving our adapted version of 1.36, one obtains that in such a universe there are two solutions: $\Delta \sim \text{const.}$ or $\Delta \sim e^{-2Ht}$: in other words, the presence of a cosmological constant kills any opportunity to form galaxies.

1.6 Power spectra

In order to make a quantitative comparison between the theories of structure and galaxy formation and the observed universe, we need to quantify the spectrum of density perturbations in the universe and relate it to the fluctuations that we discussed above, so to treat such a problem we have to work in Fourier space, and so describing these fluctuations in terms of their power spectra [4]. We'll begin by relating the spatial correlation function $\xi(r)$, to its power spectrum $P(k)$.

1.6.1 Matter power spectrum

Spatial correlation function

The spatial correlation function describes the clustering properties of galaxies in three dimensions with respect to any other galaxy, and if we write it in terms of the probability of finding pairs of galaxies, one at position \mathbf{x} , at distance r , we get

$$dN_{\text{pair}} = n_0^2 [1 + \xi(r)] dV_1 dV_2, \quad (1.39)$$

where $dN_{\text{pair}} = \langle dN_1, dN_2 \rangle$ represents the mean number of galaxy pairs in the space interval dV_1 and dV_2 , and n_0 is the mean number density of galaxies. $\xi(r)$ is observed to be well represented by a power law of the form

$$\xi(r) = \left(\frac{r}{r_0} \right)^{-\gamma}, \quad (1.40)$$

for scales between 0.01 and $\sim 10 h^{-1}$ Mpc. Here $\gamma = 1.7 - 1.8$, and $r_0 = 5 h^{-1}$ Mpc. The latter marks the scale at which the density of galaxies is twice that of the

background, as we can see from 1.39. It can be roughly interpreted as the scale below which the perturbations are non linear at the present time.

$\xi(r)$ is directly related to the density contrast $\Delta(\mathbf{x})$, that we introduced in section 1.5, as

$$\xi(r) = \langle \Delta(\mathbf{x}), \Delta(\mathbf{x} + \mathbf{r}) \rangle. \quad (1.41)$$

The perturbation spectrum

As known from classical Fourier analysis, the correlation function of a signal is the Fourier transform of its power spectrum. With a three-dimensional distribution of matter we are forced to work in terms of three-dimensional Fourier transforms; let us define the Fourier transform pair for Δ :

$$\Delta_{\mathbf{k}} = \frac{1}{V} \int_V d^3x \Delta(\mathbf{r}) e^{i\mathbf{k}\cdot\mathbf{r}} \quad ; \quad \Delta(\mathbf{r}) = \frac{V}{(2\pi)^3} \int_{V_k} d^3k \Delta_{\mathbf{k}} e^{-i\mathbf{k}\cdot\mathbf{r}}. \quad (1.42)$$

If we use Parseval theorem we can relate the integral of the squares of Δ and its Fourier transform $\Delta_{\mathbf{k}}$,

$$\frac{1}{V} \int_V d^3x \Delta^2(\mathbf{r}) = \frac{V}{(2\pi)^3} \int_{V_k} d^3k |\Delta_{\mathbf{k}}|^2, \quad (1.43)$$

where $|\Delta_{\mathbf{k}}|^2$ is the power spectrum of the fluctuation, and we will indicate it as $P(k)$; the left-hand side is simply $\langle \Delta^2 \rangle$. Now we relate $\xi(r)$ to $\langle \Delta^2 \rangle$: in this way we can find an explicit expression that links $\xi(r)$ with its power spectrum $\Delta_{\mathbf{k}}$. We have to decompose $\Delta(\mathbf{r})$ into Fourier series, and then we substitute into equation 1.41. After we noticed that all the cross terms vanish except for those for which $\mathbf{k} = \mathbf{k}'$, we can convert the Fourier series into a Fourier integral; we obtain

$$\xi(r) = \frac{V}{(2\pi)^3} \int_{V_k} d^3k |\Delta_{\mathbf{k}}|^2 e^{i\mathbf{k}\cdot\mathbf{r}}. \quad (1.44)$$

We know that $\xi(r)$ is a real function: so we are interested only in the integration over the real part of $e^{i\mathbf{k}\cdot\mathbf{r}}$, that is $\cos(kr \cos \theta)$, and we know that $\xi(r)$ is spherically symmetric, so we want to integrate over an isotropic probability distribution of θ on a sphere: the last step is to use 1.43 to write $\Delta_{\mathbf{k}} = P(k)$ in terms of an integral of $\xi(r)$; we finally obtain

$$P(k) = \frac{1}{V} \int_0^{+\infty} dr \xi(r) \frac{\sin kr}{kr} 4\pi r^2 \quad (1.45)$$

Primordial power spectrum

The primordial power spectrum describes initial fluctuations, which are those of the Inflation period: so, this power spectrum encodes the physics of the early universe. Following Dodelson's approach [13], the primordial power spectrum is described, in such a inflation model, in terms of the spectral index n_s and an amplitude of the perturbations A_s as

$$\frac{k^3}{2\pi} P_{prim}(k) = A_s \left(\frac{k}{k_{piv}} \right)^{n_s - 1}, \quad (1.46)$$

where $k_{piv} = 0.05$ Mpc is a pivot scale where the amplitude is A_s . This power-law form suggests that the fluctuation must have been very broad with no preferred scales [4]. Planck CMB measurements [7] give a value for $n_s \simeq 0.965$, which is a good confirm of Harrison-Zeldovich's hypothesis, and agrees with the most accepted versions of inflation. Power spectrum evolves following the transfer function, as we will see later.

1.6.2 CMB power spectrum

The CMB power spectrum describes the temperature fluctuations of the CMB. Also in this case we follow Dodelson's approach [13], thus we write the temperature of the CMB radiation in each point of the sky, characterized by a direction vector \hat{n} :

$$T(\hat{n}) = \sum_{\ell m} a_{\ell m} Y_{\ell m}(\hat{n}). \quad (1.47)$$

In fact, we describe the intensity of the CMB in the sky as a function of position, defined by \hat{n} , which depends on angular coordinates. We can describe temperature fluctuations because we have already said that the CMB fits a black body spectrum, and so we can make a parallelism between intensity and temperature fluctuations. This is a representation where we have decomposed $T(\hat{n})$ into spherical harmonics $Y_{\ell m}$ where the $a_{\ell m}$ are amplitudes of the expansion in this orthonormal base in spherical coordinates. Analogously to the Fourier 3D basis, the CMB power spectrum is defined as the expectation value of the square of $a_{\ell m}$:

$$C_\ell \equiv \langle |a_{\ell m}^2| \rangle = \langle a_{\ell m} a_{\ell m}^* \rangle, \quad (1.48)$$

where we have omitted the subscript m , due the isotropy of the universe [14]. In order to get to the power spectrum, we need to know $T(\hat{n}, a = 1)$, the temperature field we observe around us today, and we can do it in terms of the multipole moments of temperature fluctuations, $\Theta_{\ell m}(k, a)$. If we define for convenience the variable $\Theta(\hat{\mathbf{n}}) = \frac{\Delta T}{T}$, we can express the multipole moments as:

$$\Theta_{\ell m} = \int d\hat{\mathbf{n}} Y_{\ell m}^*(\hat{\mathbf{n}}) \Theta(\hat{\mathbf{n}}), \quad (1.49)$$

and the power spectrum as

$$\langle \Theta_{\ell m}^*, \Theta_{\ell' m'} \rangle = \delta_{\ell \ell'} \delta_{mm'} C_\ell. \quad (1.50)$$

We can then link the C_ℓ to $P(k)$ using the Zaldarriaga and Seljak approach [15]:

$$C_\ell = \frac{2}{\pi} \int dk k^2 P_{prim}(k) |\mathcal{T}_\ell(k)|^2, \quad (1.51)$$

where we have defined

$$\mathcal{T}_\ell(k) = \frac{\Theta_{\ell m}(k, a)}{\mathcal{R}(k)}, \quad (1.52)$$

and \mathcal{R} will be defined in the next section. So, we have deduced a theoretical form for CMB power spectrum.

1.7 Matter transfer function

We introduce the matter transfer function, the main object that we will investigate in this thesis: $T(k)$ describes how the shape of the initial power spectrum is modified by different physical processes, and encodes the scale dependence of $P(k)$.

More in general, we can argue that the primordial curvature fluctuations $\mathcal{R}(k)$, or equivalently $\Phi_{\tau=0}(k, z)$ where τ is the conformal time, are related to gravitational potential at late times by means of $T(k)$, for the scale dependence, and by means of $D(z)$, the growth factor that describes the time dependent growth of the matter contrast Δ at late times [13], for their time evolution. Thus we write

$$\Phi(k, z) \propto \mathcal{R}(k)T(k)D(z). \quad (1.53)$$

At late time we can relate Φ to Δ , and so to $P(k, z)$:

$$P(k, z) \propto k^{n_s} D^2(z) T^2(k). \quad (1.54)$$

So, for a fixed redshift z the power spectrum is given by

$$P(k) \propto k^{n_s} T^2(k), \quad (1.55)$$

the crucial point is the fact that $P(k)$ is fully determined by $T(k)$: so, if we know how the transfer function behaves, *i.e.* its analytic expression, it will be straightforward to compute the linear matter power spectrum. In 1986 Bardeen, Bond, Kaiser, and Szalay made a first attempt to fit $T(k)$ considering radiation, baryons, and CDM [16]: their formula, often referred as BBKS formula, is accurate up to 10%, which is too imprecise in comparison to present data [17]; we report its explicit expression in appendix A. A second attempt was made in 1998 by Eisenstein and Hu [18]: EH formula has a precision around 1%, but its expression is made up by 30 different and complicated expressions: we report its full version in appendix B. Considering the results achieved and in the absence of valid alternatives, the EH and BBKS formulas were and remain milestones in literature. We will often refer to $T(k)$ only as transfer function instead of matter transfer function.

1.8 Einstein-Boltzmann equations

Cosmology is, essentially, an application of GR coupled with statistical mechanics [13]. Consequently, nearly all cosmological findings can be derived from the integration of two pivotal equations: the Einstein equations governing gravity, see equations 1.1 and the Boltzmann equations [19] from statistical mechanics that describe matter and radiation.

Parameter estimation is one of the principal problems in modern cosmology: by this we mean the ability to relate the property of the model with observational data. An essential component in this problem is the precise calculation of various observational quantities based on cosmological models. This is done with Einstein-Boltzmann solvers, *i.e.* codes that solve the linearized Einstein-Boltzmann equations on an expanding background [20]. The history of EB solvers

is tied to the success of modern theoretical cosmology: many EB codes have been developed. Of these, **CAMB** [21] and **CLASS** [22] are kept up to date and are, by far, the most widely used as cosmological analysis tools. In order to compute the spectra for our work, we decided to use **CLASS** due to its flexibility and velocity.

Chapter 2

Machine learning

Machine Learning (ML) is a branch of Artificial Intelligence (AI) that focuses on the development of algorithms and models that enable computers to learn and make predictions or decisions based on data. Instead of being explicitly programmed, machines learn from data patterns.

In numerous scientific fields, the central goal is to construct a representation of how a collection of measurable factors (inputs) correlates with another set of factors linked to them (outputs). Once such a mathematical representation has been identified, it is possible to estimate the values of the variables of interest by performing inference procedures on experimental data. Regrettably, numerous real-world phenomena prove exceedingly intricate to depict directly as a straightforward input-output connection in a closed form. ML offers methodologies that can autonomously construct a computational model for these intricate relationships by analyzing the accessible data and optimizing a performance criterion that is specific to the problem at hand.

ML is usually divided into two main types: supervised and unsupervised learning[23]. In the predictive or supervised learning approach, the goal is to learn a mapping from inputs \mathbf{x} to outputs y , given a labeled set of input-output pairs

$$D = \{(\mathbf{x}_i, y_i)\}_{i=1}^N. \quad (2.1)$$

Here D is called the training set and N is the number of training samples. In general, however, it could be a complex structured object, such as an image, a sentence, a graph. Likewise, the format of the output or response variable can, in theory, take on various forms, but many techniques operate under the assumption that y_i falls into one of two main categories. Firstly, y_i is treated as a categorical or nominal variable originating from a finite set, where $y_i \in \{1, \dots, M\}$. This scenario is recognized as classification or pattern recognition. Secondly, y_i is regarded as a real-valued scalar, as seen in cases like income levels. In this context, the problem is referred to as regression.

The other primary category of ML is referred to as the descriptive or unsupervised learning approach. In this context, we only have access to inputs, denoted as

$$D = \{(\mathbf{x}_i)\}_{i=1}^N, \quad (2.2)$$

and the objective is to uncover patterns within the data. It is a considerably less structured problem because we are not provided with specific instructions on the types of patterns we should be seeking, and there is no straightforward error metric to employ, unlike in supervised learning, where we can measure how well our predictions of y for a given \mathbf{x} match the actual observed values.

There is a third type of ML, called reinforcement learning, that involves an agent learning to make a series of decisions in an environment to maximize a total reward. In this process, the agent interacts with the environment, takes actions to achieve a particular objective, and receives feedback in the form of rewards or penalties based on its actions. The goal is to develop a strategy that allows the agent to make decisions leading to the highest possible cumulative reward over time. This approach finds applications in fields such as robotics, game playing, and autonomous systems.

2.1 Neural networks

Neural networks have gained significant prominence and have become a dominant approach in many ML tasks, particularly in deep learning. They have demonstrated remarkable success in areas such as image recognition, natural language processing, and reinforcement learning.

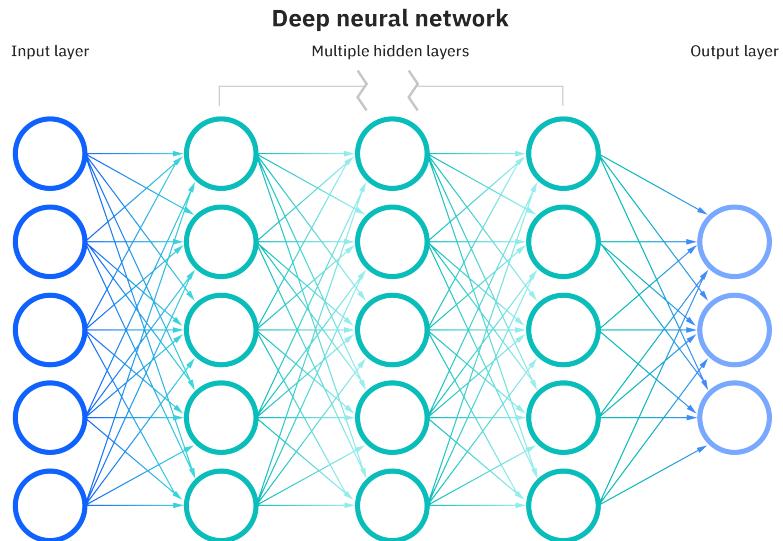


Figure 2.1: Batch diagram of a deep neural network. Credits: [24]

The simplest definition of a neural network is a nonlinear function of many variables that depends on many parameters [25]. The reason beyond the choice of a nonlinear function is that it makes it possible for the model to generalize or adapt with variety of data and to differentiate between the output: in fact, linearity limits expressiveness.

In figure 2.1 we can see a graphic representation of the structure of a neural network. But how it works? The fundamental objects of a neural network are the neurons. These objects are distributed over layers and hidden layers and are interconnected each other; "layer" refers generically to a group of neurons in a neural network, an "hidden layer" is a specific type of layer located between the input and the output of the neural network. The output of one neuron is a nonlinear function f of weighted sum of inputs:

$$f(z) \quad \text{with} \quad z = \sum_j w_j x_j + b, \quad (2.3)$$

where w_j are the so-called weights and b is the bias. The sum is extended $n \leq N$ elements, where N is the number of neuron in the previous layer. Common choices for the activation function are the sigmoid $s(z)$,

$$s(z) = \frac{1}{1 + e^{-z}}, \quad (2.4)$$

or ReLU (Rectified Linear Unit) $R(z)$,

$$R(z) = \max(0, z). \quad (2.5)$$

The sigmoid is a function that maps any input value to a value between 0 and 1, making it useful for binary classification and logistic regression problems. The shape of the sigmoid function is often referred to as an "S" shape, as it starts with a slow increase, rapidly approaches 1, and finally levels off. The function is monotonic but the derivative of the function is not, and it can happen that a neural network with a sigmoid as activation function get stuck in the training process. On the other hand, although ReLU function and its derivative both are monotonic, it maps all the negative values in zero immediately which decreases the ability of the model to fit or train from the data properly. That means any negative input given to the ReLU activation function turns the value into zero immediately in the graph, which in turns affects the resulting graph by not mapping the negative values appropriately. So, the best activation function can not be determined *a priori*, but it depends on the problem that we want to solve.

Each connection has a weight w each neuron has an offset b and each neuron has a activation function f . The values of input layer neurons are fed into the network from the user, so from outside the network. More in general, if we label with k a generic input neuron and j an output neuron, we can rewrite 2.3 with matrix formalism as

$$z_j = \sum_k w_{jk} x_k + b_j \longrightarrow z = wx + b, \quad (2.6)$$

in this way we can describe the behavior of a fixed neuron. We report an important result on neural networks, demonstrated by George Cybenko in 1989, known as 'Universal approximation theorem'[26]: any arbitrary (smooth) function, with vector input and vector output, can be approximated as well as desired by a neural network with a single hidden layer, as long as we allow for sufficiently many neurons.

Training the network

A step of the outmost importance is the training, *i.e.* the procedure used to adapt the NN weights to get it closer to the desired behaviour. This is essentially a fitting problem: we have to adjust weights in order to adapt the output function of our network $F_w(\mathbf{x})$ to training samples, that are our known data. Now, we have

$$\mathbf{y}^{out} = F_w(\mathbf{x}), \quad (2.7)$$

but we want, if it is possible

$$\mathbf{y}^{real} \simeq F(\mathbf{x}). \quad (2.8)$$

Where w also stands for biases, and F is the desired target function. This situation leads us to introduce a cost function $C(w)$, that is a measure of how well a model performs by quantifying the difference between predicted and actual outputs:

$$C(w) = \frac{1}{2} \langle ||F_w(\mathbf{x}) - F(\mathbf{x})||^2 \rangle, \quad (2.9)$$

where we average over all samples the vector norm of the difference between F_w and F . If we have N samples, we can write an approximate version of $C(w)$:

$$C(w) \approx \frac{1}{2N} \sum_{s=1}^N ||F_w(\mathbf{x}^{(s)}) - F(\mathbf{x}^{(s)})||^2, \quad (2.10)$$

the best neural network F_w is the one that minimizes the cost function. With this choice of cost function C the problem is essentially a least square fitting.

Gradient descent

Gradient descent is a optimization technique used in ML and neural network training. The cost function in gradient descent measures how well the model is doing, and the goal is to minimize this function. The model keeps tweaking its parameters until the cost function is as low as possible, indicating high accuracy.

At this moment, a problem arises: evaluating C would mean averaging over all training samples, and it is computationally heavy. In fact, the update step is defined by

$$w_j \rightarrow w_j - \eta \frac{\partial C(w)}{\partial w_j}, \quad (2.11)$$

where η is defined as step-size parameter: with a small learning rate it is likely that the algorithm remain stuck in a local minimum of C , on the other hand with a big learning rate we would not be able to find with precision the minimum of C [25], as we can see in figure 2.2:

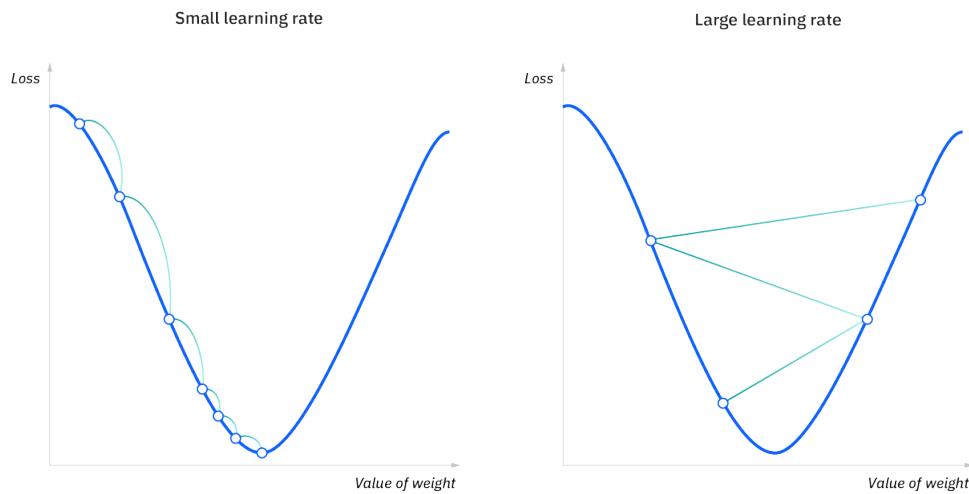


Figure 2.2: Graphic representation of gradient descent: differences between small and large learning rate. Credits: [24]

If the number of data points in our training set becomes large, we have to sum the cost of all m examples just to perform one update step. The solution to this problem is average over a few samples, and use an approximate C , as we have seen in 2.10: this is called mini batching. Another way to improve the velocity of this algorithm is using a stochastic approach: the main idea here is that instead of examining every training example to figure out how to move towards the lowest point, we can make progress by looking at just one example at a time and following its direction. This approach, called Stochastic Gradient Descent (SGD), is faster than the traditional Gradient Descent, but it takes a less precise path to reach the minimum. This is because in each step, instead of calculating the exact slope, it estimates it, which can introduce some randomness or noise into the path to convergence. So in this case we are penalizing precision in favor of speed; a good practice is to take different samples in each step.

Backpropagation

The most common use of gradient descent in a neural network is with backpropagation. This is a iterative algorithm that calculate how much each neuron's output contributed to the error with the gradient. The algorithm updates the parameters of the model, weights and biases, in the opposite direction of the gradient. We can do a physics parallelism for this backpropagation process: let us represent our neural network as a system of massive particle linked with ropes. If we imagine to pull the output layer with a mechanical force, we can see that some ropes will shrink and relax, while other ones will expand and stiffen: this is how the system evolves and changes its state, and so our neural network adjusts its weights. We do not have to confuse backpropagation with gradient descent: backpropagation plays the role of calculating the gradient, while gradient descent plays the role of

descending through the gradient.

Let us explain how backpropagation works:

- $y_j^{(n)}$ is the value of neuron j in layer n ,
- $z_j^{(n)}$ is the input value for $f(z)$, that we defined in 2.3,
- $w_{jk}^{(n,n-1)}$ is the weight that represent the connection for neuron k in layer $n-1$ feeding into neuron j in layer n .

So, from the relation 2.11 we understand that the goal is to evaluate $\frac{\partial C(w)}{\partial w_*}$, where w_* is some weight (or bias), somewhere in the net. For a particular \mathbf{x} input we can write:

$$\frac{\partial C(w, \mathbf{x})}{\partial w_*} = \sum_j (y_j^{(n)} - F_j(\mathbf{x})) \frac{\partial y_j^{(n)}}{\partial w_*} = \sum_j (y_j^{(n)} - F_j(\mathbf{x})) f'(z_j^{(n)}) \frac{\partial z_j^{(n)}}{\partial w_*}, \quad (2.12)$$

where we used that $y_j^{(n)} = f'(z_j^{(n)})$, and F_j is the target function at layer n . What we want is the change of neuron j in layer n due to change of some arbitrary weight w_* . If we apply chain rule repeatedly, we obtain

$$\frac{\partial z_j^{(n)}}{\partial w_*} = \sum_k \frac{\partial z_j^{(n)}}{\partial y_k^{(n-1)}} \frac{\partial y_k^{(n-1)}}{\partial w_*} = \sum_k w_{jk}^{(n,n-1)} f'(z_k^{(n-1)}) \frac{\partial z_k^{(n-1)}}{\partial w_*}. \quad (2.13)$$

We proceed by recursion; let us notice that each pair of layers $[n, n-1]$ contributes to multiplication with the following matrix:

$$M_{jk}^{(n,n-1)} = w_{jk}^{(n,n-1)} f'(z_k^{(n-1)}). \quad (2.14)$$

Going down the network is equivalent to do the product of these matrices. When we encounter the weight with respect to which we wanted to calculate the derivative of the cost function, we obtain

$$\frac{\partial z_k^{(\tilde{n})}}{\partial w_*} = \begin{cases} y_k^{(\tilde{n}-1)} & \text{if } w_* = w_{jk}^{(\tilde{n}, \tilde{n}-1)} \\ 1 & \text{if } w_* = b_j^{(\tilde{n})} \end{cases}, \quad (2.15)$$

where \tilde{n} is the layer where the weight that we are considering is located. Now in order to evaluate $\frac{\partial C(w)}{\partial w_*}$, we have to construct the vector

$$\Delta_j = (y_j^{(n)} - F_j(\mathbf{x})) f'(z_j^{(n)}) \quad (2.16)$$

for the output layer n , and then multiply with the M matrices 2.14 from the right. We remind that j is the index where this particular weight appear. So in each layer we have

$$\frac{\partial C(w, \mathbf{x})}{\partial w_*} = \Delta_j \frac{\partial z_j^{(n)}}{\partial w_*}, \quad (2.17)$$

and if we average over samples we obtain

$$\frac{\partial C(w)}{\partial w_*} = \begin{cases} \langle \Delta_j y_k^{(\tilde{n}-1)} \rangle & \text{if } w_* = w_{jk}^{(\tilde{n}, \tilde{n}-1)} \\ \langle \Delta_j \rangle & \text{if } w_* = b_j^{(\tilde{n})} \end{cases} . \quad (2.18)$$

This is a efficient algorithm: in fact, one single backpropagation pass through the network yields all the derivatives of C with respect to all the weights and biases. This brings a huge advantage compared to doing all the derivatives numerically for each weight.

2.2 Symbolic Regression

Physical laws are often expressed by simple laws, such as Newton's or Coulomb's law. However, there are situations where it is not possible to find an easy mathematical expression. In this context we introduce another ML technique: Symbolic Regression (SR).

Symbolic regression seeks the space of mathematical expressions that best fits a given dataset [27]; it describes a supervised learning task where the model space is spanned by analytic expressions [28]. The advantage over a classic neural network is basically that symbolic regression gives an interpretable model instead of the black-box output of a neural network. We refer to regression as an approach for modelling the relationship between a response and one or more explanatory variables.

In a classic regression the functional form of the equation is determined *a priori* by the user: the best known example is linear regression, but we can mention also polynomial and logistic regressions. This fact penalizes the flexibility of the model, as is limited to the functional form we chose. A classic regression is applied in situation in which we have a strong theoretical knowledge of the problem that we are studying (*i.e.* we know the functional formula). For example we can apply linear regression if we want calculate sound speed c_s , assuming we have measured wavelengths λ_i and we know frequencies ν_i , because we know that $c_s = \lambda_i \nu_i$.

On the other hand, in a symbolic regression, relax the assumptions on the functional formula that describes our phenomenon. This feature results in high flexibility, and with this approach we can discover complex relationship that best fit our data. In this case the algorithmic approach is based on gradient descent, that we have already explained, and evolutionary algorithms: so we can look for our solution among many potential models. Sometimes it is difficult to interpret the output, because we can find ourselves in a situation where the combination of our input variables are complicated. So, we shall apply symbolic regression in situation where we have a unknown relationship between input data and output, and complex and non-linear patterns [29][30]. The principal drawback is the risk to overfit, that is when a model learns the training data too well, capturing noise and random fluctuations rather than just the underlying patterns, and this fact leads to a loss of predictive power: when the algorithm overfits, it is very hard to scale the relations found. In general, the best approach in this situations is to use

a combination of symbolic regression and neural networks, as is shown in [31]. In general, it is useful to use constraints on the model, this can greatly reduce the complexity of the output [32].

2.2.1 Genetic algorithms

The traditional symbolic regression approach is based on Genetic Algorithms (GA), because it provides a large space for possible analytical equations [27]. A genetic algorithm is a ML process based on evolutionary biology, where a population of individuals evolves to better adapt to a certain environment: in this case we have a population of mathematical expressions that evolve to adapt to a certain dataset provided by the user. This evolutionary mechanism adapts well to many heavy computational problems, or to task with a high number of possible solutions. In those situation it is useful to have an adaptive computer program, *i.e.* that performs well in a changing environment [33]. All this features can be satisfied by a code that emulates biological evolution: in fact evolution is a method that permits us to explore a very large amount of possibilities and solutions. An important thing to emphasize is that the adaptation criteria continually changes, this is meaning that evolution explore often a different set of possibilities: this is a useful feature for an adaptive program. In order to show how a GA works, let us provide a simple example taken from [32]:

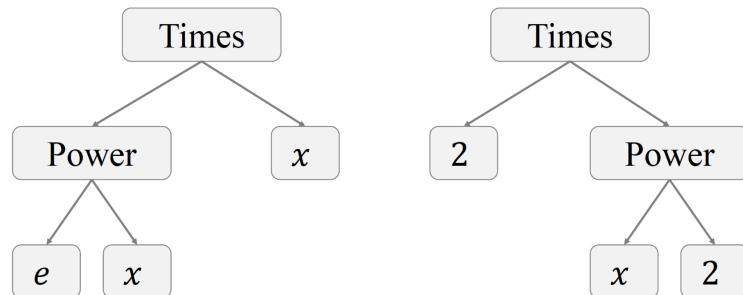


Figure 2.3: Two exemplifying grammatical expressions corresponding to the individuals xe^x and $2x^2$.

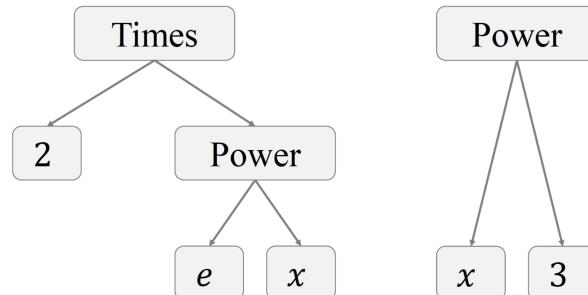


Figure 2.4: The two selected individuals have been combined to produce two new individuals: $2e^x$ and x^3 .

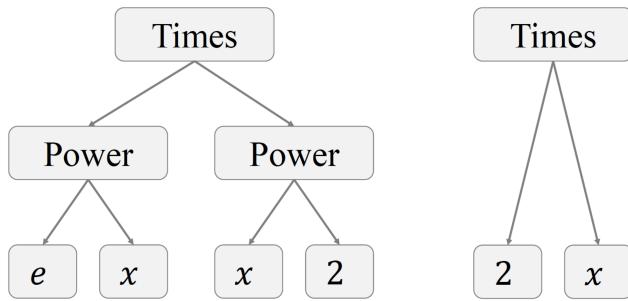


Figure 2.5: The two selected individuals have suffered a mutation to become two new individuals: x^2e^x and $2x$.

in figure 2.3, there are two expressions in tree representation, *i.e.*, two individuals xe^x and $2x^2$. Let us assume that these individuals were selected for reproduction. Figure 2.4 shows possible combinations of the two individuals, while figure 2.5 shows their possible mutations. For instance, the new individual in the left-hand side of figure 2.4 is born from the combination of the expressions e^x and 2 from its parents, while the other individual is made from the remaining parts, x and x^2 , using the proper basic operation “times” or product. During the mutation stage, a selected individual is altered randomly. As shown in figure 2.5, for instance, the individual xe^x mutated to x^2e^x , and the individual $2x^2$ mutated to $2x$.

2.3 PySR

PySR is an open source library for symbolic regression [28]; it is a evolutionary genetic algorithm that uses multiple evolving populations to solve problems, and these populations evolve independently at their own pace. PySR is a software tool that relies on a distributed backend, uses a versatile search method.

The core process in PySR, which runs independently for each population, follows a traditional evolutionary algorithm approach. It employs tournament selection to choose individuals and incorporates various mutation and crossover techniques to create new individuals. This main algorithm can be represented as a evolve-simplify-optimize loop, designed to optimize unknown scalar constants in the discovered expression. Let us focus on this three steps: "Evolve" is like a continuous cycle of selection-based evolution. It applies changes to a set of equations for a specific number of times. This process is similar to how species evolve in nature through natural selection. "Simplify" refers to making equations simpler. During the process, equations are transformed into equivalent forms using algebraic simplification rules. This step is done occasionally to make the search for equations more efficient but does not limit the potential discoveries. "Optimize" involves using a mathematical optimization technique (usually BFGS [34], but other optimizers can be used) to fine-tune the constants in the equations. This step is crucial for improving the accuracy and practical use of the discovered equations, especially when dealing with equations that contain real numerical

values.

The flexibility of this library can be understood from the additional features that are provided: let us explain briefly some of these. If we face noisy data, PySR supports three useful features: first, an optional denoising preprocessing step that takes into account a Gaussian, a white and a constant noise. Second, we can specify a set of weights for input data points, for example it is useful when we know the uncertainties on the data. Third, we can define a custom likelihood that best adapts the problem we are facing. This third feature is very useful in general, not only with noisy data, because permits us to define arbitrary loss function: in this way we can define arbitrary regression objectives and classification-based losses. Another powerful tool is the ability to define custom operators: various branches of science have specialized functions and formulas that serve distinct purposes within their respective fields. These formulas often have specific meanings and applications. The last feature that we mention is the possibility to specify constraints, such as dimensional constraints: we have already said that one drawback of symbolic regression is the difficulty in interpreting the output. With the use of the constraints we can reduce the complexity tree of the algorithm and this will lead us to a more interpretable output.

2.3.1 Tests on PySR limits

Before applying the symbolic regression algorithms to the problems we want to solve, we are interested in understanding the potential and limitations of our code, in particular of the library we are going to use for regression, PySR. This is because if we analyse data in different ranges, or with a more or less dense sampling, we can obtain results with greater or lesser accuracy, depending on the context. This fact does not surprise us, it is something that is also present in nature: for example, if we limit ourselves within a certain regime we can describe reality with Newtonian mechanics, and in this context it also works very well. To study the universe in all its complexity, this model is no longer sufficient and relativity must be introduced; even in ultra-precision physical measurements we need to introduce relativity. The most constructive example in this respect is the measurement of perihelion precession of Mercury [35], which although it is an effect on a solar scale, thus in the range in which we assume Newtonian mechanics to be valid, is predicted by relativity and not by mechanics.

For our tests, we decided to use a simple function of this form

$$x^2 + 2.5382 \cos(x) - 0.5. \quad (2.19)$$

We want to analyse how the algorithm behaves in relation to the conditions in which we decide to set ourselves to solve the regression problem.

One dimensional case

As a first case, we chose to use this function as $f : \mathbb{R} \rightarrow \mathbb{R}$, so that it can be visualized in a plot. We then generated an input dataset x consisting of 100 randomly drawn points from a normal distribution with mean $\mu = 0$ and standard

deviation $\sigma = 5$. We then used equation 2.19 to obtain a dataset y on which to apply the algorithm; we also created two other datasets by adding noise to y , in the first case uniform, *i.e.* that comes from a uniform distribution ($\mu = 1$) and in the second case Gaussian ($\mu = 0, \sigma = 2$). We present in figure 2.6 a graphical representation of this situation.

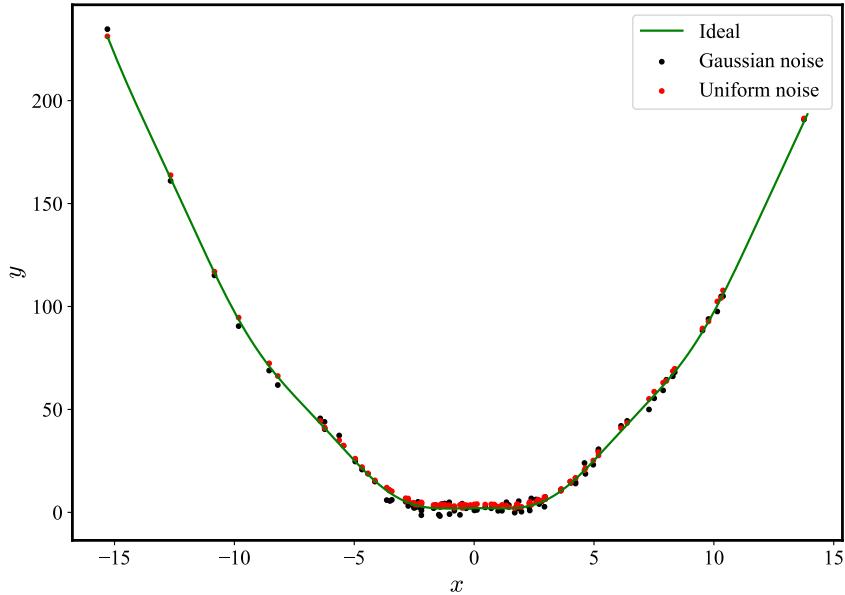


Figure 2.6: Graphical representation of the one-dimensional data set generated using equation 2.19, to which we have added a Gaussian noise in one case and a uniform noise in the other.

At this point we specified the working space for the symbolic regression algorithm:

1. we arbitrarily set the number of cycles, or evolutionary epochs, to 40;
2. we defined a set of basic operators, the grammar of the genetic algorithm, with which it will evolve the expressions found;
3. as a last step we defined a loss function, which the algorithm will try to minimize to find the best expression that fits data: we choose 3.6.

For now, given the simplicity of the problem, we do not need to specify much more.

With this configuration, the code manages to recover optimally, as expected, the analytical formula searched for 2.19, using the dataset y . Since the noise is uniform and has an average of 1, the model will attempt to capture this random component of the data. This can lead the model to compensate for the additional noise and fit the data so that it appears as if it has an offset equal to 1 compared

to the true underlying model, *i.e.* the ideal model. Indeed, Gaussian noise has a symmetric distribution around zero, so on average the noise will have a zero effect on the data. However, because the regression model tries to fit the data as well as possible, it can compensate for some of this noise, resulting in an offset from the true underlying model: we expect a value of the offset near 0.5. These limitations may compromise the search for better analytical expression. We report in the table 2.1 the numerical values of what has been discussed so far: the offset is defined as $f(x) - y$, where $f(x)$ is the symbolic equation found by our algorithm.

noise	expression	offset
/	$x^2 + 2.54 \cos(x) - 0.50$	$3.14 \cdot 10^{-7}$
Uniform	$x^2 + 2.51 \cos(x) + 0.35$	0.84
Gaussian	$x^2 + 2.82 \cos(x)$	0.49

Table 2.1: Comparison of ideal situation and addition of noise for one-dimensional dataset with 100 data points

Multi dimensional case

In this second case we created a dataset \mathbf{x} with 100 points, with 5 features each, that lives in \mathbb{R}^5 . We wanted a $g : \mathbb{R}^2 \rightarrow \mathbb{R}$, but we also want to understand if the algorithm can choose the right data from a larger set. So we modified equation 2.19 to get:

$$x_0^2 + 2.5382 \cos(x_3) - 0.5, \quad (2.20)$$

where x_0 and x_3 are respectively the first and the fourth component of \mathbf{x} .

noise	expression	offset
/	$x_0^2 + 2.53 \cos(x_3) - 0.502$	$-2.44 \cdot 10^{-3}$
uniform	$x_0^2 + 2.60 \cos(x_3) + 0.518$	1.02
Gaussian	$x_0^2 + \cos(x_3)$	0.38

Table 2.2: Comparison of ideal situation and addition of noise for multi-dimensional dataset with 100 data points

We created 3 output datasets as in the one-dimensional case, with the same uniform and Gaussian noise. We have obtained results very similar to the previous case, which we report in table 2.2. But now we want to take a step further: we would like to understand if, by increasing the points or decreasing the noise, we can recover the correct form of equation 2.20 even in the case of noise.

In the first test we decided to increase the number of points in the same interval as before, to see if, with a denser distribution, we can improve the results of the program on the datasets with noise. The result was negative, in fact we found expressions substantially identical to previous ones, where we considered 100 points. So we have, always with 1000 points, lowered the magnitude of the

noise, switching to a normal distribution with $\mu = 0$ and $\sigma = 0.5$, and a uniform distribution with $\mu = 0.5$: in these conditions we finally recovered expressions similar to equation 2.20. We report this results in table 2.3. The results of these tests evidences that PySR has some difficulties with big noises: it is fair to mention that there is a specific denoising feature implemented in this library [28], which can be very helpful in situations like this. However, due to the lack of noise in the data that CLASS will retrieve, which is noise-free, its application is beyond our scope.

noise	expression	offset
/	$x_0^2 + 2.52 \cos(x_3) - 0.499$	$2 \cdot 10^{-3}$
Uniform	$x_0^2 + 2.52 \cos(x_3) - 0.255$	0.24
Gaussian	$x_0^2 + 2.52 \cos(x_3) - 0.571$	-0.07

Table 2.3: Comparison of ideal situation and addition of a lower noise than in the previous case for multi-dimensional dataset with 1000 data points

Chapter 3

Application of symbolic regression to $T(k)$

As we already said, modern cosmology is strongly determined by the precision with which the cosmological parameters are known, and those are related to how sharply we solve Einstein-Boltzmann equations [36]. The objective of this thesis work is to try to understand if, and if yes how well, symbolic regression algorithms can find an analytical formula for the description of functions and spectra that nowadays we know only as numerical solution of Einstein-Boltzmann equations. In this chapter we will focus on the transfer function $T(k)$ of the matter power spectrum $P(k, z)$, a key function as part of the study of matter density fluctuations in the universe, that are encodes in $P(k, z)$. The principle with which we start this work is similar to a position of a scientist like Kepler: we start from data and nothing else, trying to limit to the maximum the *a priori* hypotheses.

3.1 Dependence of $P(k)$ on cosmological parameters

First, we study the dependence of the matter power spectrum on the parameters of the Λ CDM model, which we have reported in table 1.1, to understand their effect on the spectrum shape. We have to mention that our study is done at a fixed value of redshift, $z = 0$, so we will omit its dependence.

So, regarding $P(k)$, we are interested to study its dependence on the density of baryonic matter ω_b , the density of cold dark matter ω_c and the neutrino mass m_ν . To do this we varied these three parameters one at a time, fixing the others to the fiducial values, assumed to be those from the Planck mission [7]. As we can see in figures 3.1 3.2 and 3.3. Evident oscillations, called Baryonic Acoustic Oscillations (BAO) in the region around $k = 0.1h$ Mpc are seen. Before recombination occurred, within the cosmic plasma electrons and protons were fully coupled to radiation via electromagnetic forces. The photon pressure countered the gravitational collapse of baryonic fluctuations, resulting in oscillations within the photon-baryon mixture. Such oscillations left their mark on the matter power spectrum: their presence is crucial to determine the composition of the

early universe [37].

3.1.1 Baryonic matter

In order to determine the dependence of $P(k)$ on ω_b , we have fixed ω_c and m_ν and chose $\omega_b \in [0.007, 0.042]$, that is, bracketing the Planck reference value ($\omega_b = 0.0223828$).

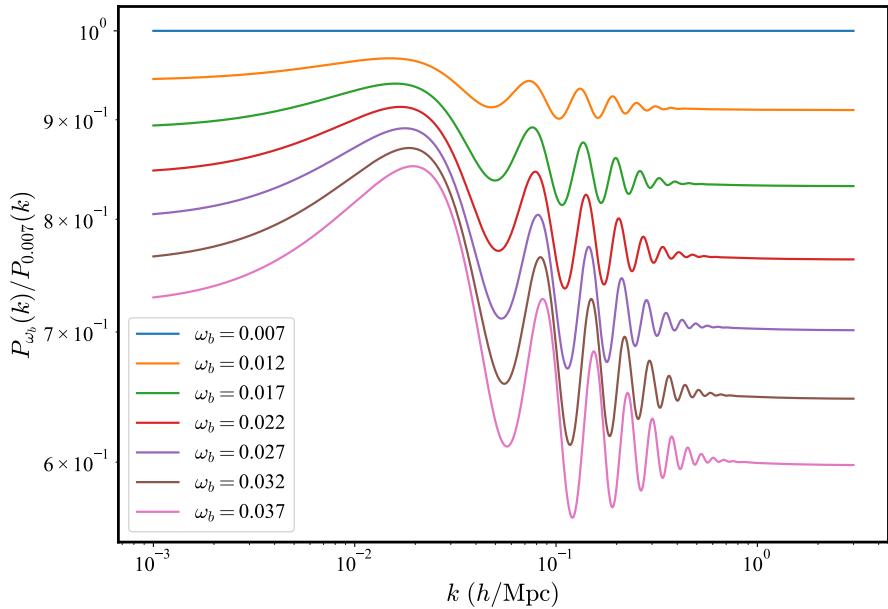


Figure 3.1: The ratio of $P(k)$ in different cosmologies (described varying the parameter ω_b) to that in the fiducial cosmology, *i.e.* for $\omega_b = 0.007$.

From figure 3.1, we see that the amplitude of the oscillations, which we can describe by a value A_{BAO} , increases as a function of ω_b . We can try to model this dependence as:

$$A_{BAO} \propto (\omega_b)^\alpha , \quad \alpha > 0. \quad (3.1)$$

Thus, in a high-baryon universe, we have strong BAO features.

3.1.2 Cold dark matter

To study the dependence of the spectrum on ω_c , we fix ω_b and m_ν ; the range we choose in this case is $\omega_c \in [0.06, 0.18]$, following the same reasoning as above: the reference value is in this case $\omega_c = 0.1801075$.

In this regime, as can be seen from the figure 3.2 we obtain

$$A_{BAO} \propto (\omega_c)^\beta , \quad \beta < 0, \quad (3.2)$$

i.e. in a universe with a strong presence of cold dark matter we obtain a flattening of these oscillations, due to their dependence on the baryonic fraction, reported in equation B.28.

Reality tells us, on the basis of the most recent measurements, that the most correct answer is the one in which ω_b and ω_c are considered to be shared: therefore in determining our spectra we will use the fiducial Planck values which are those extrapolated from the measurements we have already discussed.

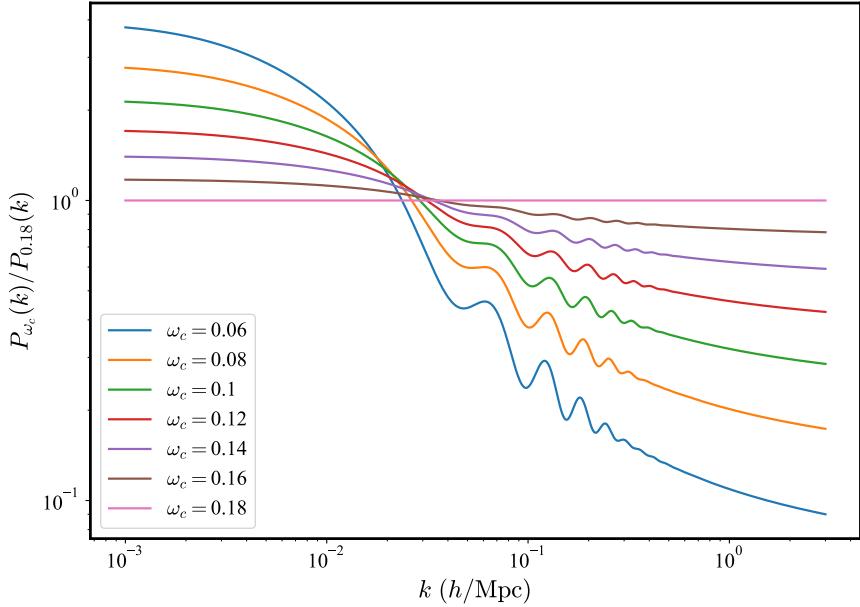


Figure 3.2: The ratio of $P(k)$ in different cosmologies (described varying the parameter ω_c) to that in the fiducial cosmology, *i.e.* for $\omega_c = 0.18$.

3.1.3 Massive neutrinos

As a third and final evaluation, we fix ω_b and ω_c , letting m_ν vary in the range $0.06 \text{ eV} \leq \sum_\nu m_\nu \leq 0.12 \text{ eV}$. In this case, the lower bound for m_ν corresponds to the minimum mass allowed by neutrino flavour oscillation experiments [38] [39], while the upper bound is chosen to be that provided by Planck [7].

What we obtain, looking at figure 3.3 is

$$P(k) \propto (m_\nu)^\gamma \quad , \quad \gamma < 0, \quad (3.3)$$

i.e. at sufficiently small scales, free streaming massive neutrinos imprint their effects on the cosmological evolution, producing a suppression of the matter power spectrum. We will assume for the determination of the spectra $m_\nu = 0.06 \text{ eV}$ because this is the value that best fits the experimental data.

Now, let us first divide the work into two main parts: in the first part we search a fitting formula as a function of the density of baryons and matter, *i.e.*

considering mass-less neutrinos. In the second part we will consider that neutrinos actually have a finite rest mass: we already seen in section 3.1.3 that this fact have an impact on the shape of $T(k)$.

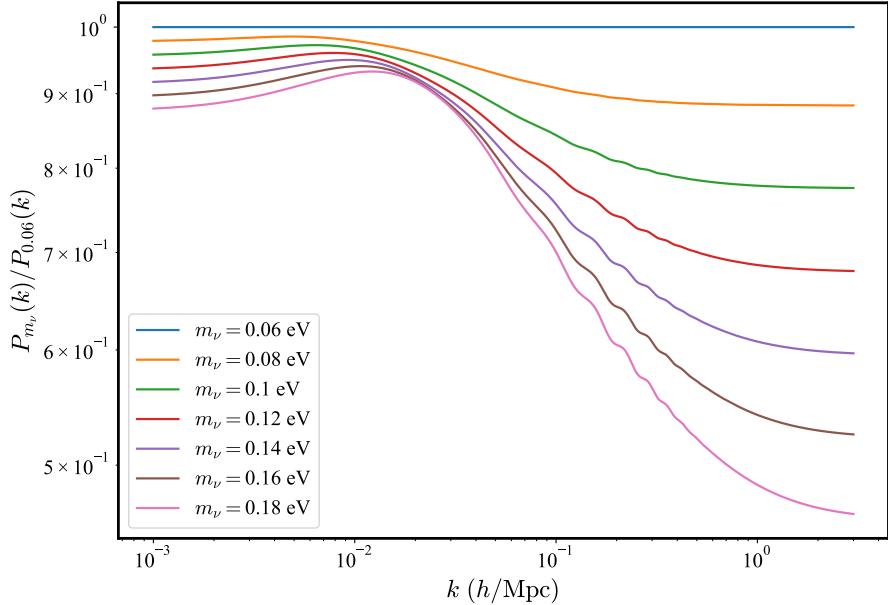


Figure 3.3: The ratio of $P(k)$ in different cosmologies (described varying the parameter m_ν) to that in the fiducial cosmology, *i.e.* for $m_\nu = 0.06$ eV.

3.2 Data

3.2.1 Baryons and CDM

We use the Boltzmann solver **CLASS**[22] to calculate the set of matter transfer functions $T(k)$ that we will use for training our algorithm. Within this computational framework, each pair of parameters $\{\omega_b, \omega_m\}$ defines a unique cosmological scenario, for which we can compute $T(k)$ as a function of the wavenumber k at a fixed redshift $z = 0$. To investigate the influence of these parameters on the transfer function, we computed them over a grid comprising 4×4 pairs of $\{\omega_b, \omega_m\}$. We restricted the parameter ranges to $\omega_b \in [0.0214, 0.0234]$ and $\omega_m \in [0.13, 0.15]$, values that are approximately within a 10σ interval around the best-fit parameters of by Planck [7]. For each of the 16 cosmological scenarios under consideration, the **CLASS** software extracts a set of 114 data points denoted as $\{k, T\}$. So, our first dataset comprises a total of 1824 data points, organized in rows represented as $\{k, \omega_b, \omega_m, T\}$.

3.2.2 Baryons, CDM and massive neutrinos

In addition to baryonic matter and cold dark matter, massive neutrinos can have a detectable effect on the shape of $T(k)$: to incorporate this influence of massive neutrinos on the matter transfer function, we employ the computational tool **CLASS**. Specifically, we calculate the transfer function $T(k)$ as a function of the parameters ω_b , ω_m within the same parameter ranges defined in the previous section. We now use the parameter ω_ν , introduced in equation 1.23, which represents the contribution of a single massive neutrino. We assume the total mass of massive neutrinos to fall within the interval $0.06 \text{ eV} \leq \sum_\nu m_\nu \leq 0.12 \text{ eV}$, which corresponds to the range discussed in Section 3.1.3.

In this context, we construct a grid consisting of $4 \times 4 \times 4$ data points, encompassing the parameters $\{\omega_b, \omega_m, \omega_\nu\}$ and compute the transfer function for each triplet. For each of the 64 cosmological scenarios **CLASS** yields 114 data points, denoted as $\{k, T\}$. Consequently, our final dataset takes the form of a table with dimensions 7296×5 , with individual data points represented as $\{k, \omega_b, \omega_m, \omega_\nu, T\}$.

3.3 Performance evaluation metrics

We assess the adequacy of the fit of a given analytical expression through the utilization of two distinct metrics. The first one is the Mean Absolute Percentage Error (MAPE)

$$\text{MAPE} = \frac{100}{N} \sum_{i=1}^N \left| \frac{T_{i,\text{CLASS}} - T_{i,\text{PySR}}}{T_{i,\text{CLASS}}} \right|. \quad (3.4)$$

The second metric is the Root Mean Square Error (RMSE)

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (T_{i,\text{CLASS}} - T_{i,\text{PySR}})^2}. \quad (3.5)$$

In the context of our dataset, N represents the number of the data points, while T_i is a condensed representation denoting the evaluation of the transfer function at the specific values $\{k_i, \omega_{b_i}, \omega_{m_i}, \omega_{\nu_i}\}$. We choose to employ MAPE and RMSE because those are widely accepted measures in the field of regression analysis due to their interpretability and well-established mathematical properties. MAPE expresses errors as a percentage of the observed values, making it intuitive and readily understandable, while RMSE provides a measure of the standard deviation of the residuals. As is the norm, we establish as the data acceptance criterion a MAPE of less than 5%.

3.4 Fitting formulae from PySR

In this section, we provide our fitting equations for the transfer function. We also share some information about the specific configuration we used for symbolic

regression. The code below contains the specific settings incorporated into the regression model during one of our search steps, which we will explain later.

```

1 model = pysrRegressor(
2     model_selection="best",
3     niterations=100000, # < Increase me for better results
4     binary_operators=[ "+", "*", "- ", "/ ", "^" ],
5     unary_operators=[ 'log10' ],
6     constraints={ '^': (5, 2) },
7     nested_constraints={ "^^": { "^^": 2 } },
8     maxsize=20,
9     loss="loss(prediction, target) = ((prediction -
10       target)^2) / (target^2)",
11     # ^ Custom loss function (julia syntax)
12     turbo=True,
13 )

```

Code 3.1: Regression model in PySR

`PySSRRegressor` is the `PySR` module that serves as the tool for defining the attributes and parameters of the symbolic regression model. For further details on the parameters used, please refer to the official documentation [40]. A key starting point for our approach was not to seek a solution with *a priori* form, but rather to allow the algorithm to discover the best possible form without any external assistance: we seek to investigate if and how `PySR` can determine the relationship between the transfer function and cosmological parameters. We would like to emphasize the necessity of adhering to three essential mathematical and physical criteria when we will consider our proposed solutions:

1. The transfer function has to have specific limits: $T \rightarrow 0$ when $k \rightarrow \infty$, and $T \rightarrow 1$ when $k \rightarrow 0$,
2. the transfer function exclusively assumes non-negative values,
3. the transfer function is a smooth function, *i.e.* $T(k) \in C^\infty(\mathbb{R}^+, \mathbb{R}^+)$.

3.4.1 Baryons and CDM

For our investigation we used different approaches which are presented hereafter.

No external hints on input variables

In our initial methodology applied to the dataset presented in section 3.2.1, our approach involved inputting three variables $\{k, \omega_b, \omega_c\}$, along with the expression of T , computed by `CLASS`. For the purpose of optimizing our model, we opted for employing the mean squared error as loss function, *i.e.*

$$\text{loss} = (T_{i,\text{CLASS}} - T_{i,\text{PySR}})^2, \quad (3.6)$$

which quantifies the relative error. In the context of these specific circumstances, the code has derived a function as follows:

$$T(k, \omega_m) = \frac{1}{1 + 11 \left(\frac{k}{\omega_m} \right)^{1.38}}. \quad (3.7)$$

Analyzing the value of the metrics, we observe that the MAPE exceeds the 5% threshold, albeit by a slight margin. However, upon closer examination of the graphical representation of the error, reported in 3.4, it becomes evident that the predictive accuracy of the algorithm decreases significantly at smaller scales, with a maximum percentage error of 62.81%, primarily attributable to the low values assumed within that range. Consequently, the point-to-point discrepancies are relatively low in absolute terms, but non in a relative perspective.

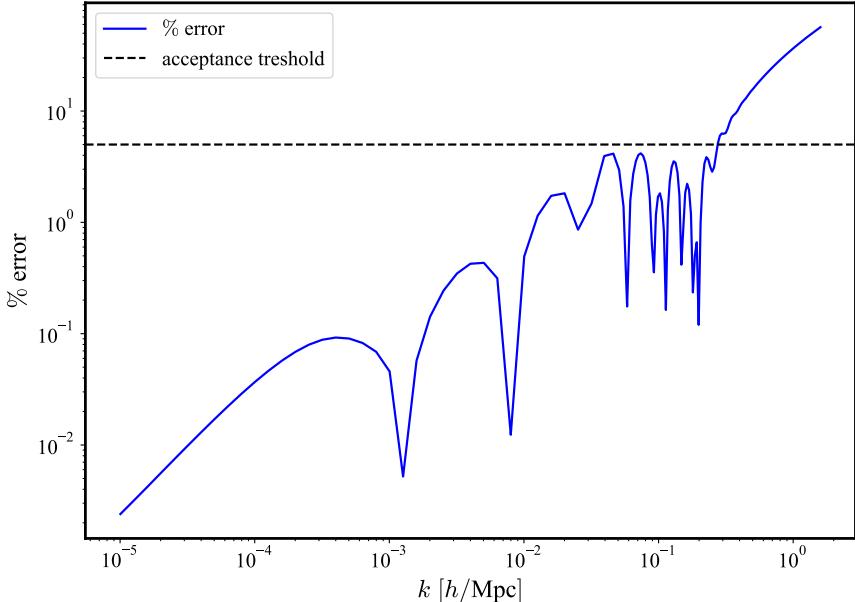


Figure 3.4: Percentage error of the fitting formula as a function of the scale k for fixed $\{\omega_b, \omega_m\} = \{0.0214, 0.13\}$.

The RMSE value has been omitted from our discussion due to its inherent nature as a direct derivative of the loss function. As expected, it assumes a substantially low value, consistent with the internal backpropagation mechanism of the PySR framework. Consequently, we are inclined to pursue an alternative approach in light of these observations.

A hint from BBKS formula

Indeed, the form of the variable closely resembles the form of BBKS formula, as documented in A.1. Given that genetic algorithms tend to perform more efficiently

when dealing with fewer input variables, we have decided to introduce a new input variable x that follows a similar format:

$$x = \frac{k \text{ Mpc}}{\omega_m - \omega_b}. \quad (3.8)$$

This strategic choice aligns with the principle of optimizing the input space for the most effective utilization of GAs in our research context. When examining the output under these revised initial conditions, we obtain this symbolic expression:

$$T(x) = \frac{1}{1 + 8.88x^{1.40}}. \quad (3.9)$$

In this situation, we observe an enhancement in the results. Specifically, we achieve a MAPE of 3.84%, with the maximum percentage error decreasing to 44.87%. Nonetheless, the issue at large k , as previously described, persists, even if to a lesser extent. Therefore, it is advisable to explore alternative approaches for improving our model while retaining the variable x that we have introduced.

Using logarithms

As previously mentioned, the transfer function is constrained to positive values. To address the issue encountered at small scales, we have considered performing regression on the logarithm of the transfer function rather than on T itself. This approach is chosen because taking the logarithm of numbers between 0 and 1 results in a significant spread of values and an increase in their absolute magnitudes. Consequently, this enables us to recover higher accuracy at small scales. It is worth noting that the algorithm correctly identified that we were analyzing the logarithm of T , in fact the expression derived in this case is:

$$T(x) = \frac{1}{1 + 6.60x + 3.14x \log(x) + 1.87x^2}, \quad (3.10)$$

where we have reported directly $T(x)$ instead of $\log(T(k))$. The results of the metrics are as follows: MAPE of 1.29% and a maximum percentage error of 4.26%. Despite the goodness of fit, we encounter some shortcomings concerning the physical and mathematical requirements. Specifically, this function is not defined at 0, it is not strictly monotonically decreasing, and it assumes values, even if slightly, greater than 1 as x approaches 0. In light of these observations, we explored an alternative approach by modifying the loss function.

Using χ^2 as loss function

The second approach employed to address the small-scale problem involved a modification of the loss function, transitioning from 3.6 to

$$\text{loss} = \frac{(T_{i,\text{CLASS}} - T_{i,\text{PySR}})^2}{(T_{i,\text{CLASS}})^2}. \quad (3.11)$$

This alteration reflects a shift in focus from relative error to absolute error. The rationale behind this change is that, with an aim to enhance accuracy at low values, this new loss function should prove to be a favorable choice. The outcomes of our experimentation indeed validate our initial hypothesis. Specifically, we derive a symbolic expression of this form:

$$T(x) = \frac{1}{1 + 4.24x^{1.08} + 4.24x^{1.74}}. \quad (3.12)$$

To bolster our confidence in the results, let us delve into the performance metrics. The MAPE stands at 1.51%, the RMSE is 0.01, and the maximum percentage error peaks at 6.08%. Additionally, it is worth noting that the mathematical constraints set forth earlier are upheld by the function under consideration.

For a visual representation of our findings, we have plotted the results in figure 3.5, a graphical illustration where we observe a good alignment with the empirical data. Hence, this represents our ultimate fitting formula in the context of our research focused on cosmological models that incorporate massless neutrinos.

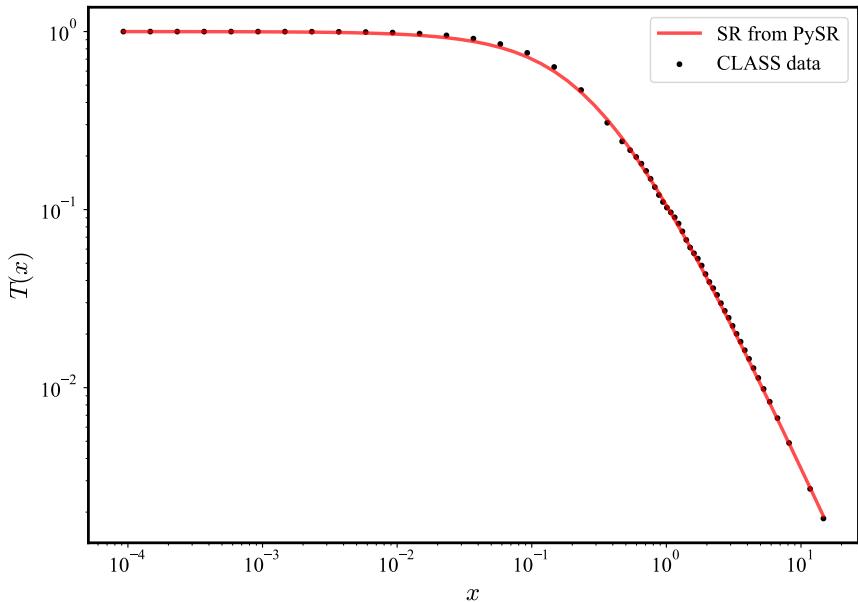


Figure 3.5: Comparison between CLASS data (dots) and the expression found by PySR (solid line) for fixed $\{\omega_b, \omega_m\} = \{0.0214, 0.13\}$.

3.4.2 Baryons, CDM and massive neutrinos

As we previously said, it is essential to acknowledge that neutrinos possess a finite mass, thereby imparting a small, but non-negligible contribution to the overall mass density of the universe, as we have seen in section 3.1.3. In alignment with the method employed in the preceding section, we introduce a variable q denoted

as

$$q = \frac{k \text{ Mpc}}{\omega_m - \omega_b + \omega_\nu}, \quad (3.13)$$

which accounts for the influence of massive neutrinos. Once again, we have to say that this particular approach yielded the most favorable outcome. To elucidate further, we adopted the chi-square 3.11 as our chosen loss function, and subsequently executed a regression analysis using the dataset presented in 3.2.2. As a result, we report our derived fitting formula as follows:

$$T(q) = \frac{1}{1 + 7.06q^{1.28} + 1.69q^2}. \quad (3.14)$$

Upon a brief examination of the analysis, we have achieved a MAPE of 1.40%, a RMSE of 0.005, and a maximum percentage error of 6.68%. These outcomes represent an improvement when contrasted with the scenario involving mass-less neutrinos. Importantly, the solution obtained adheres to the mathematical conditions mandated for the transfer function. In the following, we also present a graphical depiction of our findings:

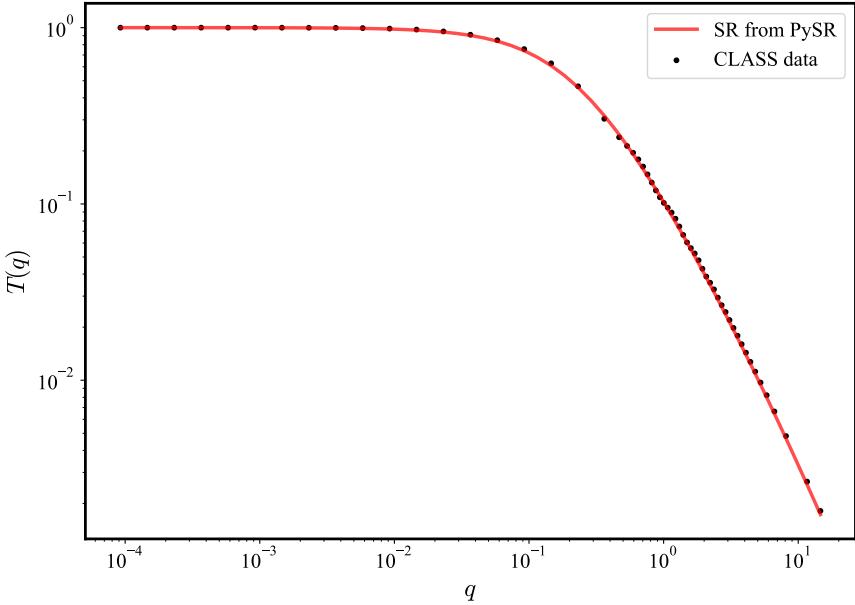


Figure 3.6: Comparison between CLASS data (dots) and the expression found by PySR (solid line) for fixed $\{\omega_b, \omega_m, \omega_\nu\} = \{0.0214, 0.13, 6.38 \cdot 10^{-4}\}$.

3.5 Comparison with BBKS and EH formulae

We will now conduct a analysis of the results exploring the domains where the identified models manifest suboptimal performance, coupled with an elucidation

of the underlying causal factors. Furthermore, our objective is to draw a comparative assessment between these models and two well-established formulations, specifically the BBKS and EH formulae. In both instances, our observations indicate that, whether dealing with neutrinos characterized by zero mass or massive counterparts, all three formulae (our proposed formulation, BBKS, and EH) demonstrate commendable levels of precision when scrutinized at larger scales, defined as those with wavenumbers k less than $0.01h$ Mpc. Conversely, as we look smaller scales, characterized by k values exceeding $0.01h$ Mpc, we discern a decline in accuracy, attributed to the non-negligible influence of the baryonic component. This phenomenon, commonly referred to as Baryonic Acoustic Oscillations, was explained in section 3.1.1. Now we present a tabulated comparison of the accuracy exhibited by the three formulae in the mass-less neutrinos scenario:

formula	expression	error
PySR	equation 3.12	1.51%
BBKS	appendix A	8.71%
EH	appendix B.1	0.78%

Table 3.1: Expression and MAPE of the fitting formulae for the matter transfer function as a function of k, ω_b and ω_m .

We see that in this situation EH formula is the best among all: let us notice that even if our formula is slightly less accurate than EH, it is much simpler than EH, as can be seen in appendix B. In table 3.2 we compare of the precision of our formula and EH one, in the case of massive neutrinos:

formula	expression	error
PySR	equation 3.14	1.40%
EH	appendix B.2	1.23%

Table 3.2: Expression and MAPE of the fitting formulae for the matter transfer function as a function of k, ω_b, ω_m and $\omega\nu$.

This scenario mirrors the previous one, even if with a noteworthy observation: now the precision of the two formulae are of the same order of magnitude, and so this convergence, coupled with the inherent simplicity of our expression, underscores the robust utility of symbolic regression within this specific context.

3.6 Validation test

As final step we want to understand how well our SR algorithm scales with increasing complexity, extrapolating outside the parameter ranges used in the training. This test can also reveal if the algorithm is able to generalize to bigger datasets and

it is useful in evaluating the robustness of the code, revealing problems and drawbacks in our symbolic formulae. To this end, we have created two new dataset: for mass-less neutrinos we have generated a new 4×4 grid comprising 16 pairs of $\{\omega_b, \omega_m\}$, and calculate $T(k)$ for each pair. For massive neutrinos we created a $4 \times 4 \times 4$ grid comprising 64 triplets of $\{\omega_b, \omega_m, \omega_\nu\}$. In this case we have considered the intervals $\omega_b \in [0.0234, 0.0254]$, $\omega_m \in [0.15, 0.17]$ and $m_\nu \in [0.12, 0.18]$ eV, because our purpose is to test symbolic expressions on unknown parameters. Tables 3.3 and 3.4 compare the errors of training and extrapolation data:

mass-less neutrinos	parameters range	error	standard deviation
training	section 3.2.1	1.51%	$9.2 \cdot 10^{-3}$
extrapolation	this section	1.30%	$8.6 \cdot 10^{-3}$

Table 3.3: Expression, MAPE and RMSE of PySR fitting formulae for the matter transfer function in a cosmology scenario with mass-less neutrinos as a function of k and different sets of $\{\omega_b, \omega_m\}$.

massive neutrinos	parameters range	error	standard deviation
training	section 3.2.2	1.40%	$4.7 \cdot 10^{-3}$
extrapolation	this section	1.40%	$5.0 \cdot 10^{-3}$

Table 3.4: Expression, MAPE and RMSE of PySR fitting formulae for the matter transfer function in a cosmology scenario with massive neutrinos as a function of k and different sets of $\{\omega_b, \omega_m, \omega_\nu\}$.

We can see that all indicators have essentially preserved the same values. Even more so in the case of mass-less neutrinos in which they even improved. These preliminary results seem to show that the model has generalized outside its training range, demonstrating its ability to extrapolate.

Chapter 4

Application of symbolic regression to C_ℓ^{EE}

In this chapter we present the results obtained from the application of symbolic regression to the power spectrum of CMB polarization anisotropies, namely C_ℓ^{EE} . The CMB radiation power spectrum is important because it identifies those small fluctuations present in the temperature of the CMB, which are the progenitors of the structures that we now observe in the universe, such as galaxies. We have a strict link between temperature and density fluctuations: if we imagine that the early universe was a uniform fluid, a warmer zone may have been denser than a colder zone. So, in a nutshell, anisotropies of CMB temperature are a photography of primordial density fluctuations. We applied the same approach we have adopted for $T(k)$, *i.e.* we used PySR to find a symbolic representation of this spectrum at large scales. The goal is to identify a symbolic representation that can be effectively employed in the preprocessing of data to enhance the precision of C_ℓ analyses.

4.1 Dependence of CMB power spectrum on τ_{reio}

The CMB Power Spectrum can be broken down into three main components: TT , TE and EE . These three components reflect different correlations between temperature (T) and polarisation (E) fluctuations in the cosmic background radiation. Our study will mainly focus on the EE component, C_ℓ^{EE} , which is the one describing the auto-correlation of polarisation fluctuations.

Polarisation fluctuations in the CMB refer to variations in the polarisation direction of the cosmic background radiation. These variations are caused by fluctuations in the initial conditions of the early universe, particularly during the period of recombination, when electrons and protons combined to form neutral atoms, allowing the radiation to spread freely through the universe. We are in particular interested in the dependence of C_ℓ^{EE} on τ_{reio} , the optical depth to reionization. It is a unitless quantity which provides a measure of the line-of-sight free-electron opacity to CMB radiation.

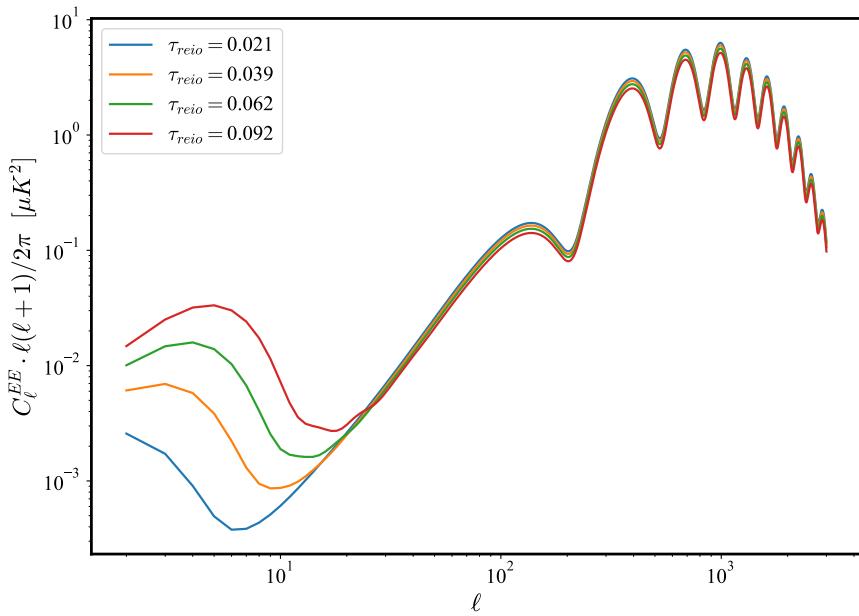


Figure 4.1: The power spectrum of polarization anisotropies, C_ℓ^{EE} for different values of τ_{reio} , plotted on a logarithmic scale to emphasise the direct proportionality for $\ell \ll 30$.

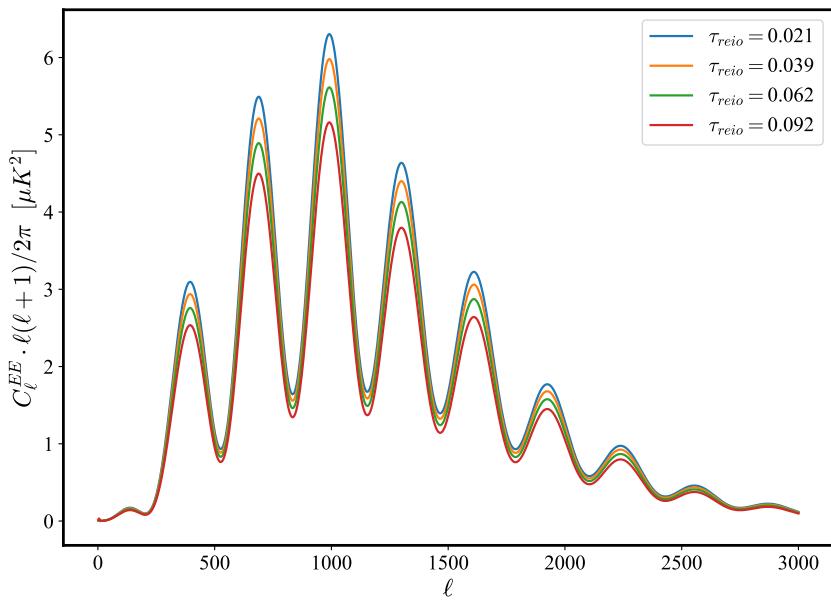


Figure 4.2: The C_ℓ^{EE} for different values of τ_{reio} , plotted on a linear scale to emphasise the inverse proportionality for $\ell \gg 30$.

Under the assumption of instantaneous, complete reionization at redshift z_{reio} ,

τ_{reio} is computed as

$$\tau_{reio} = \int_0^{z_{reio}} dl n_e \sigma_T, \quad (4.1)$$

i.e. the integral of the electron density times the Thomson cross section over the geometrical path length computed between $z = 0$ and z_{reio} . Assuming a fixed dependence of electron density on redshift, a larger value of τ_{reio} implies a larger value of z_{reio} and thus an earlier onset of star and galaxy formation; $\tau_{reio} = 0$ implies no reionization at all.

Therefore, to evaluate the dependence on τ_{reio} , we first generated a set of 20 spectra corresponding to 20 values of τ_{reio} , covering a $\pm 5\sigma$ interval around the Planck value $\tau_{reio} = 0.0561$. We then computed the ratio of C_ℓ^{EE} at τ_{reio} to that in the fiducial cosmology, i.e. the one with $\tau_{reio}=0.0206$. As we can see from the figures 4.1 4.2 the proportionality with respect to τ_{reio} is not always the same: from figure 4.1 we see that for small ℓ (i.e. at large scales) C_ℓ^{EE} is directly proportional to τ_{reio} , while on small scales, for $\ell >> 30$, figure 4.2 shows an inverse proportionality. More precisely, it is known from the literature [41] [42] that the following relations hold:

$$C_\ell^{EE} \propto e^{-2\tau} \quad , \quad \ell >> 30, \quad (4.2)$$

$$C_\ell^{EE} \propto \tau^2 \quad , \quad \ell << 30. \quad (4.3)$$

Here we will focus on the latter, large-scale regime.

4.2 Data

Again, we used **CLASS** to get the CMB power spectrum C_ℓ . In this case we varied only the parameter τ_{reio} , and calculated C_ℓ as a function of the multipole moment ℓ , that we remind is related to the angular scale θ [39], at redshift $z = 0$. To understand the dependence on τ_{reio} , we have created a vector $v \in \mathbb{R}^{20}$ containing 20 values of this parameter; we selected a range $\tau_{reio} \in [0.0206, 0.0916]$, in which those values are approximately within a 5σ interval around the best fit parameter reported by Planck value [7]. For each cosmological scenario under consideration, **CLASS** retrieves 2998 data points of all three components, and so our initial dataset contains 59960 data points, organised in rows as follows $\{\ell, \tau_{reio}, C_\ell^{EE}\}$.

4.3 Fitting formulae from PySR

In this section, we provide our fitting equations for CMB *EE* power spectrum. We will use from now on τ instead of τ_{reio} for simplicity of notation. We are particularly interested in the behaviour of the function at large scales (small ℓ); as we already said in section 4.1, in order to emphasize the quadratic dependence of C_ℓ^{EE} from τ , described in equation 4.3, We then computed the ratio of C_ℓ^{EE}

at τ to that in the fiducial cosmology, *i.e.* the one with $\tau = \tau_{min} = 0.0206$, and we define this quantity as:

$$\mathcal{F}(\ell, \tau) = \frac{C_\ell^{EE}(\tau)}{C_\ell^{EE}(\tau_{min})}. \quad (4.4)$$

We present $\mathcal{F}(\ell, \tau)$ in figure 4.3. Our research was conducted at large scales, *i.e.* we considered $\ell \in [2, 50]$ because CLASS retrieves C_ℓ^{EE} for $\ell \geq 2$ and we choose 50 as final value because close to this value we begin to see the effect of τ dependency described in the equation 4.2.

The structure of the regressor is very similar to the one presented for the transfer function, that we have reported in code 3.1. The main difference lies in the grammar of operators, which is much broader here, and includes hyperbolic functions, trigonometric functions, logarithm and exponential function. In this case we have no analytical limits for this spectrum, as we had for the transfer function. Consequently, we can see that the ratios under consideration do not exhibit divergence at zero and exhibit behavior similar to an exponential decay $e^{-2\tau}$, tending towards infinity, as we can see in figure 4.3.

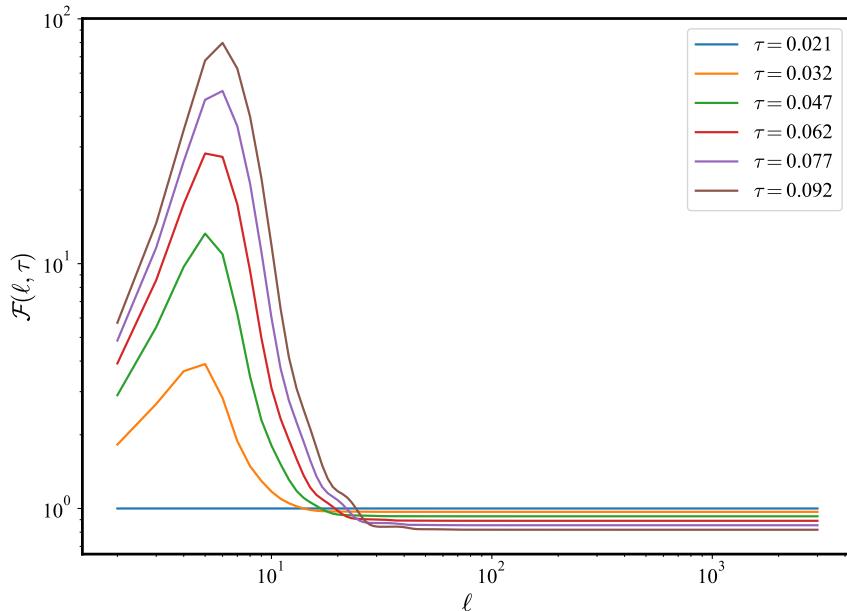


Figure 4.3: Graphical representation of $\mathcal{F}(\ell, \tau)$ as function of multipole $\ell \in [2, 3000]$ for six fixed τ values.

To study this problem we have used two distinct methodologies. First, we directly applied SR to the dataset; subsequently, we applied a preprocessing to remove the contribution associated with the exponential term $e^{-2\tau}$.

4.3.1 SR on CLASS data

The description again splits in two main parts. The first approach we have adopted is separation of variables, due to $C_\ell^{EE}(\tau)$ dependence on τ and ℓ .

Separation of variables

We aim at finding a function $\bar{\mathcal{F}}$ of the form:

$$\bar{\mathcal{F}}(\ell, \tau) = f(\ell) \cdot g(\tau), \quad (4.5)$$

for the description of the ratios at low ℓ . On the basis of the results of previous investigation on the transfer function, we decided to adopt equation 3.11 as the most appropriate loss function for our SR model.

In this context, we fixed τ to the maximum value in the interval, *i.e.* $\tau = 0.0916$ because it is the value that make the ratio bigger in the considered ℓ , namely $\ell \in [2, 50]$. In this situation the input dataset will be $\{\ell, \mathcal{F}(\ell, \tau_{min})\}$, for a total 48 data points.

The function reconstructed by the symbolic code has been the following:

$$f(\ell) = e^{\frac{\ell}{0.32(-1.39\ell - \ell) + 1.16}} - 0.124, \quad (4.6)$$

which is smooth in the interval considered. Subsequently, we fixed ℓ to 5 because we want to get an expression for the dependency on τ : we chose $\ell = 5$ because at this ℓ value we note, as we can see in figure 4.3, that the ratios have a peak. The input dataset is $\{\tau, \mathcal{F}(5, \tau)\}$, for a total of 20 data points.

The model gives this output equation:

$$g(\tau) = 8.05 \cdot 10^4 \tau^3 - 1.89 \cdot 10^3 \tau^2 + 1.11. \quad (4.7)$$

Although we have not found a perfect τ^2 proportionality, we note a similar behaviour for $\tau \rightarrow 0$. Thus, we obtain a \mathcal{F} with this form:

$$\bar{\mathcal{F}}(\ell, \tau) = [e^{\frac{\ell}{0.32(-1.39\ell - \ell) + 1.16}} - 0.124] \cdot [8.05 \cdot 10^4 \tau^3 - 1.89 \cdot 10^3 \tau^2 + 1.11]. \quad (4.8)$$

We can see the graphical results of those regressions in the following plots:

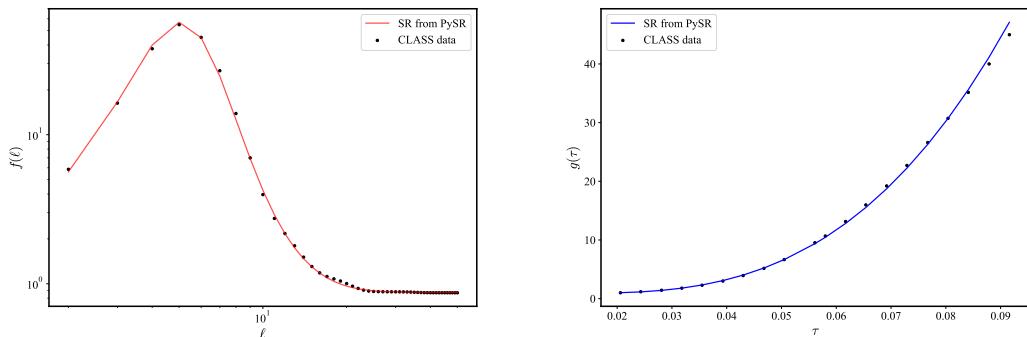


Figure 4.4: Left: comparison between CLASS data (dots) and the expression found by PySR (solid line) for dataset $\{\ell, \mathcal{F}(\ell, \tau_{min})\}$. Right: comparison between CLASS data (dots) and the expression found by PySR (solid line) for dataset $\{\tau, \mathcal{F}(5, \tau)\}$.

Examining the results of our error metric, it becomes evident that the MAPE for the function $\bar{\mathcal{F}}(\ell, \tau)$ significantly surpasses the established 5% threshold. Specifically, we computed an average MAPE of 68.10%, with the highest MAPE occurring at $\tau = 0.0206$ and reaching 88.94%. This issue originates from our implicit assumption that the shape of the peak of the ratios remains consistent for all values of τ . However, this assumption is not true. Consequently, we must abandon this approach and explore an alternative strategy.

Two-variables regression We now will consider a different approach, more direct than the previous one: we will compute a SR inputting two variables $\{\ell, \tau\}$ along with the expression for the ratios. The algorithm is now searching directly a symbolic expression for \mathcal{F} ; we have an input dataset composed by $\{\ell, \tau, \mathcal{F}(\ell, \tau)\}$, with a total of 960 data points. The best equation that our algorithm retrieved is:

$$\bar{\mathcal{F}}(\ell, \tau) = -\tau + \cosh(0.019^{\sin(2.42\tau - \log(\ell+1.25))}\tau). \quad (4.9)$$

We note that we obtained a fit with a MAPE of 9.01%, with a maximum percentage error for $\tau = 0.0916$ which stands at 16.77%. These values are above the 5% threshold that we chose as acceptance; in addition, we obtained an RMSE of 12.18, primarily attributable to the behavior of the symbolic expression as ℓ becomes large. Specifically, it tends to approach 1 for all τ , which is inconsistent with the observed data. However, it is fair to say that we have slightly improved the performance of our algorithm, which is still useful since we are seeking an expression to be used in the preprocessing of a more sophisticated analysis, so we are trying to capture the general behaviour, rather than trying to fit too accurately the expression. In any case, it is worth seeing whether a more accurate expression can be found.

4.3.2 SR on preprocessed CLASS data

As reported in equation 4.2, we observe an exponential decay relationship with respect to τ at small angular scales. So, to obtain an expression that tends to 1 when $\ell \gg 30$ we can preprocess the spectra by dividing them by $e^{-2\tau}$. Hence, we define a new quantity:

$$\mathcal{H}(\ell, \tau) = \frac{C_\ell^{EE}(\tau)}{C_\ell^{EE}(\tau_{min})} \frac{e^{-2\tau_{min}}}{e^{-2\tau}} = \mathcal{F}(\ell, \tau) \frac{e^{2\tau}}{e^{2\tau_{min}}}. \quad (4.10)$$

We want an $\mathcal{H} \rightarrow 1$ for $\ell \gg 30$ because this form provide a known limit for the studied quantity, making it easier to find a suitable solution for our algorithm. We will use only the *two-variables regression* approach, because we have demonstrated that is more convenient than the *separation of variables* approach. The input dataset is now $\{\ell, \tau, \mathcal{H}(\ell, \tau)\}$, with a total of 960 data points. We obtained as best expression the following equation:

$$\bar{\mathcal{H}}(\ell, \tau) = \cosh(54.3^{-\sin(2.77\tau - \log(\ell+1.40))}\tau). \quad (4.11)$$

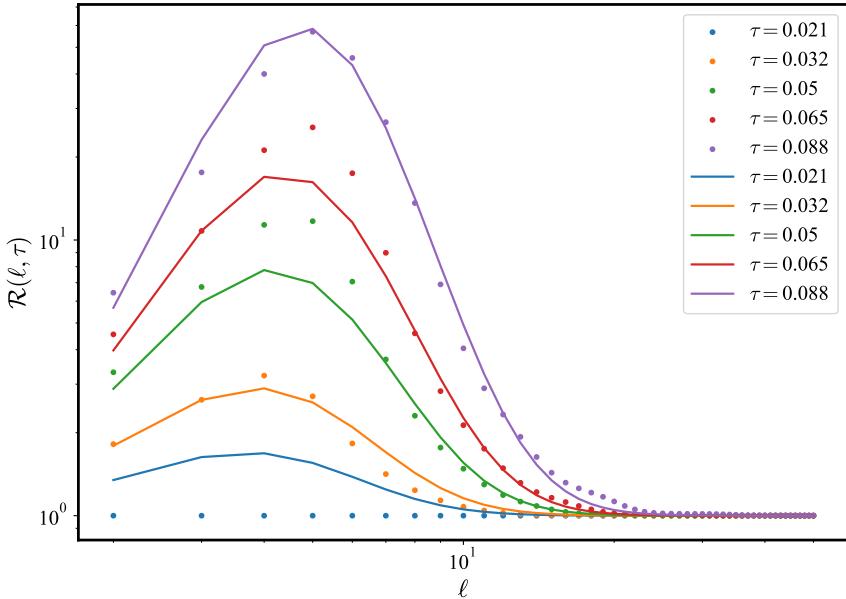


Figure 4.5: comparison between CLASS data, \mathcal{H} , (dots) and the expression found by PySR , $\bar{\mathcal{H}}$, (solid line) as function of multipole $\ell \in [2, 50]$ for five fixed τ values.

In figures 4.5 and 4.6, \mathcal{R} indicates a generic function of ℓ at fixed τ . Looking at the performance evaluation metrics, we get a MAPE of 3.64%, a RMSE of 0.06 and a maximum percentage error of 6.59% for $\tau = 0.0206$. We have reported in figure 4.5 the results of this fit. The indicators are below the acceptance threshold, and thus show a considerable improvement over the case without data preprocessing; in addition the symbolic expression respects the mathematical constraints that we have requested. We report in table 4.1 a summary of the results obtained.

	expression	error	standard deviation
$\bar{\mathcal{F}}(\ell, \tau)$	equation 4.9	9.01%	12.8
$\bar{\mathcal{H}}(\ell, \tau)$	equation 4.11	3.64%	0.06

Table 4.1: Expression, MAPE and RMSE of the PySR fitting formulae for the ratios, in the first case without and in the second case with preprocessing.

Hence, preprocessing has demonstrated its effectiveness in identifying an improved symbolic representation. This outcomes is now of greater value, making it a suitable preprocessing step for conducting a more precise analysis of C_ℓ^{EE} spectrum via neural networks algorithms [41].

4.4 Determination of a general formula for the ratios

In this last section, we want to understand how well our fitting formula can extrapolate to higher ℓ . We expect that, thanks to the preprocessing that we introduced, we will get a good compatibility to real data. If we evaluate

$$\lim_{\ell \rightarrow \infty} \bar{\mathcal{H}}(\ell, \tau) \quad (4.12)$$

for a fixed $\tau \neq 0$, we can conclude that the limit is undefined due to the fact that as the logarithm of ℓ approaches infinity, it results in evaluating the limit of $\sin(\infty)$, which exhibits oscillatory behavior.

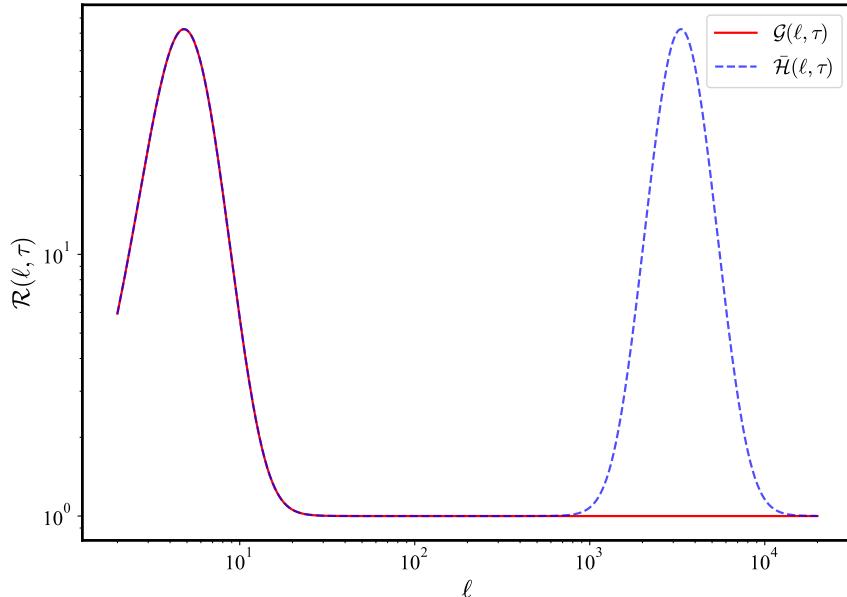


Figure 4.6: Comparison between the fitting formula $\bar{\mathcal{H}}$ and our modified solution \mathcal{G} , which reproduces the correct boundary for $\ell \rightarrow \infty$ of \mathcal{H} , avoiding the periodic peak presence that we can find in $\bar{\mathcal{H}}$.

So, this is telling us that the solution we have discovered has the potential for extension to higher values of ℓ , but it may not apply to all $\ell \in \mathbb{N}$: in fact, we know that the fixed limit for $\mathcal{H}(\ell, \tau)$ for $\ell \rightarrow \infty$ is 1, so we can represent the asymptotic behaviour with the function $w(\ell) = 1$. What we would like is to get a function that accounts for the formula we found, equation 4.11 and respects the limit to infinity, namely $w(\ell)$. The approach we have taken is to use a smooth transition function, a sigmoid, which we already encountered when we introduced NN in chapter 2, of the form

$$s_{\alpha, \beta}(\ell) = \frac{\alpha}{1 + e^{-\beta(\ell - \ell_0)}}. \quad (4.13)$$

We then have to determine the parameters α , β and ℓ_0 . To determine ℓ_0 we note that we want to switch on $w(\ell)$ and switch off $\bar{\mathcal{H}}(\ell, \tau)$ at the point where our expression loses physical meaning. To find this point we evaluated $\partial_\ell \mathcal{H}(\ell, \tau)$ and we checked that after the peak the derivative is ≤ 0 up to a certain $\bar{\ell} = \bar{\ell}(\tau)$: after this point the derivative of 4.11 takes positive values, which makes it lose sense. We identify ℓ_0 with $\bar{\ell}$, and so we obtain that now s depends also on τ , *i.e.* we have $s = s(\ell, \tau)$. Since $\mathcal{H}(\ell, \tau)$ already tends to 1 from $\ell \sim 20$ onwards, we can choose a β quite arbitrarily, in fact in the interval $\ell \in [20, \ell_0]$ we have $\mathcal{H}(\ell, \tau) \simeq w(\ell)$ and the function remains close to one until $\ell \sim 1000$: for simplicity we have chose $\beta = 1$. At this point, it's straightforward to put $\alpha = 1$ because for $\ell \rightarrow \infty$ we have $s_{\alpha,1}(\ell, \tau) \rightarrow \alpha$, and we want that our new function respects the constraint $\mathcal{H} \rightarrow 1$ for $\ell \rightarrow \infty$. In conclusion, we define

$$\mathcal{G}(\ell, \tau) = [1 - s_{1,1}(\ell, \tau)]\bar{\mathcal{H}}(\ell, \tau) + s_{1,1}(\ell, \tau)w(\ell). \quad (4.14)$$

In this way we have defined a smooth function that accurately represents the ratios $\mathcal{H}(\ell, \tau)$, avoiding the presence of periodic peaks, as we can see from figure 4.6.

Chapter 5

Conclusions

In this thesis we have applied Symbolic Regression (SR) techniques to the problem of the statistical description of the cosmological large-scale structure. In particular, we considered the modelization of two-point statistics in Fourier space, *i.e.* the power spectrum of density fluctuations. We studied power spectrum statistics in two fashions: on one side, we considered power matter spectrum $P(k)$ and on the other the power spectrum of angular anisotropies in the Cosmic Microwave background C_ℓ , in particular we focused on the power spectrum of CMB polarization anisotropies, namely C_ℓ^{EE} . These two objects do not have an analytic expression, but can only be calculated numerically by integrating Einstein-Boltzmann equations. Clearly, if an analytic expression could be found to approximate those quantities, this would speed up significantly all calculations when comparing models with observations.

To tackle this problem, we chose to use a Machine learning (ML) approach: machine learning has gained significant importance in recent years due to its ability to effectively tackle scientific problems, as we elucidated in chapter 2. Since our goal is to discover a possible new equations describing our power spectra, we decided to apply symbolic regression, an ML technique that explores the space of mathematical expressions to identify the most appropriate fit for a given dataset.

We applied SR techniques using the Python library PySR. First, we looked for an approximate analytical expression for the matter transfer function, reporting our result in chapter 3. Second, we searched for an expression for the dependency of the polarization spectrum C_ℓ^{EE} on τ_{reio} and multipole ℓ , which we indagated in chapter 4.

In the first case, we found simple analytical expressions that we compared to the Eisenstein-Hu (EH) approximations for $T(k)$. Although our expressions are more compact than the EH expression, they almost reach the same level of precision. We also checked that our expressions can extrapolate, *i.e.* that they are able to accurately fit the numerical predictions for $T(k)$ beyond their range, a considerable advantage of NN based approaches.

In the second case, we also managed to identify an analytic expression C_ℓ^{EE} spectrum on τ_{reio} and ℓ . Also in this case we checked both the accuracy and the extrapolations properties of the found expressions.

Our results substantiate the effectiveness of this symbolic regression approach,

which represents a complementary and promising tool for the approximations of the modelling of cosmological observable quantities. Future applications have the potential to solve even more complex problems.

For example, a possible development could be the refinement of the application of symbolic regression algorithms to get more precise expressions.

Another interesting case would be that of massive neutrinos cosmologies, in which we know that $T(k)$ is modified introducing a scale dependent growth factor, *i.e.* $D = D(k, z)$. It would be intriguing to apply symbolic regression to such a problem, to possibly get a more complete characterization of the matter power spectrum $P(k, z)$.

Finally, another interesting development could be to use symbolic regression as a way of pre-processing the input information to a neural network. In particular, this should be feasible for our derived C_ℓ^{EE} expression.

Appendix A

BBKS formula

The following fitting formula drawn from the work of Bardeen (1985), Bond and Szalay (1983), Bond and Efstathiou (1984), and Efstathiou and Bond (1985) accurately reproduce the transfer function at late times for universes dominated by collision-less relics of the big bang.

We consider cold dark matter and adiabatic fluctuations. We define a dimensionless variable q :

$$q \equiv \frac{k\theta^{\frac{1}{2}}}{(\omega_m - \omega_b)\text{Mpc}^{-1}}, \quad \theta \equiv \frac{\rho_r}{1.68\rho_\gamma} \quad (\text{A.1})$$

θ is a measure of the ratio of the energy density in relativistic particles (photons plus neutrinos) to that in photons. The expression $\theta = 1$ corresponds to three flavors of relativistic neutrinos plus the photons. Now we present the fitting formula for the case $\omega_b \ll \omega_m$:

$$T_{BBKS}^{(m)}(k; \omega_m, \omega_b) = \frac{\ln(1 + 2.34q)}{2.34q} [1 + 3.89q + (16.1q)^2 + (5.46q)^3 + (6.71q)^4]^{-\frac{1}{4}} \quad (\text{A.2})$$

if ω_b is not negligible, i.e. on small scales, we obtain

$$T_{BBKS}^{(m,b)}(k; \omega_m, \omega_b) = T_{BBKS}^{(m)}(k; \omega_m, \omega_b) \left[\frac{1 + (kR_{Jr})^2}{2} \right]^{-\frac{1}{2}} \quad (\text{A.3})$$

where

$$R_{Jr} \equiv 1.6(\omega_m - \omega_b)^{-\frac{1}{2}} \text{kpc} \quad (\text{A.4})$$

Appendix B

EH formulae

B.1 No massive neutrinos

The transfer function given by Eisenstein and Hu [18] in a universe in which we consider mass-less neutrinos has the following form:

$$T(k) = \frac{\Omega_b}{\Omega_0} T_b(k) + \frac{\Omega_c}{\Omega_0} T_c(k) \quad (\text{B.1})$$

Where $\Omega_0 = \Omega_b + \Omega_c$. The terms involved in this formula are the following:

$$T_b = \left[\frac{\tilde{T}_0(k, 1, 1)}{1 + (ks/5.2)^2} + \frac{\alpha_b}{1 + (\beta_b ks)^3} e^{-\left(\frac{k}{k_{Silk}}\right)^{1.4}} \right] j_0(k\tilde{s}) \quad (\text{B.2})$$

$$T_c = f\tilde{T}_0(k, 1, \beta_c) + (1 - f)\tilde{T}_0(k, \alpha_c, \beta_c) \quad (\text{B.3})$$

$$\tilde{T}_0(k, \alpha_c, \beta_c) = \frac{\ln(e + 1.8\beta_c q)}{\ln(e + 1.8\beta_c q) + Cq^2} \quad (\text{B.4})$$

$$f = \frac{1}{1 + (ks/5.4)^4}, \quad C = \frac{14.2}{\alpha_c} + \frac{386}{1 + 69.9q^{1.08}} \quad (\text{B.5})$$

$$q = \frac{k}{13.41k_{eq}}, \quad R = \frac{3\rho_b}{4\rho_\gamma} = 31.5\omega_b\Theta_{2.7}^{-2}\text{Mpc}^{-1} \quad (\text{B.6})$$

$$k_{Silk} = 1.6\omega_b^{0.52}\omega_0^{0.73} \left[1 + (10.4\omega_0)^{-0.95} \right] \text{Mpc}^{-1}, \quad k_{eq} = 7.46 \cdot 10^{-2}\omega_0\Theta_{2.7}^{-2} \text{Mpc}^{-1} \quad (\text{B.7})$$

$$\alpha_b = 2.07k_{eq}s(1 + R_d)^{-3/4}G(y(z_d)) \quad (\text{B.8})$$

$$\beta_b = 0.5 + \frac{\Omega_b}{\Omega_0} + \left(3 - 2\frac{\Omega_b}{\Omega_0} \right) \sqrt{(17.2\omega_0)^2 + 1}, \quad \beta_{node} = 8.41\omega_0^{0.435} \quad (\text{B.9})$$

$$s = \frac{2}{3k_{eq}} \sqrt{\frac{6}{R_{eq}}} \ln \left(\frac{\sqrt{1+R_d} + \sqrt{R_d+R_{eq}}}{1 + \sqrt{R_{eq}}} \right), \quad \tilde{s} = s \left[1 + \left(\frac{\beta_{node}}{ks} \right)^3 \right]^{-\frac{1}{3}} \quad (\text{B.10})$$

$$G(y) = -6y\sqrt{1+y} + y(2+3y) \ln \left(\frac{\sqrt{1+y}+1}{\sqrt{1+y}-1} \right), \quad y = y(z) = \frac{1+z_{eq}}{1+z} \quad (\text{B.11})$$

$$\alpha_c = a_1^{-\Omega_b/\Omega_0} a_2^{-(\Omega_b/\Omega_0)^3} \quad (\text{B.12})$$

$$a_1 = (46.9\omega_0)^{0.670}[1+(32.1\omega_0)^{-0.532}], \quad a_2 = (12.0\omega_0)^{0.424}[1+(45.0\omega_0)^{-0.582}] \quad (\text{B.13})$$

$$\beta_c = \left\{ 1 + b_1 \left[\left(\frac{\Omega_c}{\Omega_0} \right)^{b_2} - 1 \right] \right\}^{-1} \quad (\text{B.14})$$

$$b_1 = 0.944[1+(458\omega_0)^{-0.708}]^{-1}, \quad b_2 = (0.395\omega_0)^{-0.0266} \quad (\text{B.15})$$

$$z_{eq} = 2.50 \cdot 10^4 \omega_0 \Theta_{2.7}^{-4}, \quad z_d = 1291 \frac{\omega_0^{0.251}}{1 + 0.0659\omega_0^{0-828}} [1 + c_1\omega_b^{c_2}] \quad (\text{B.16})$$

$$c_1 = 0.313\omega_0^{-0.419}[1+0.607\omega_0^{0.674}], \quad c_2 = 0.238\omega_0^{0.223} \quad (\text{B.17})$$

where we have used that $\omega_0 = (\Omega_c + \Omega_b)h^2$, $T_{CMB} = 2.7\Theta_{2.7}$ K, $R_i = R(z_i)$, $y_i = y(z_i)$. We recommend reading [18] for the meaning of all these parameters.

B.2 Considering massive neutrinos

The transfer function given by Eisenstein and Hu [43] in a universe in which we consider massive neutrinos has the following form:

$$T_{cb\nu}(k, z) = T_{master}(k) \frac{(k, z)}{D_1(z)} \quad (\text{B.18})$$

The terms involved in this formula are the following:

$$T_{master} = T_{sup}(k)B(k) \quad (\text{B.19})$$

$$D_{cb\nu}(z, q) = \left[f_{cb}^{0.7/p_{cb}} + \left(\frac{D_1(z)}{1 + y_{fs}(q, f_\nu)} \right)^{0.7} \right]^{p_{cb}/0.7} D_1(z)^{1-p_{cb}} \quad (\text{B.20})$$

$$D_1(z) = \frac{1+z_{eq}}{1+z} \frac{5\Omega(z)}{2} \left[\Omega(z)^{4/7} - \Omega_\Lambda(z) + \left(1 + \frac{\Omega(z)}{2}\right) \left(1 + \frac{\Omega_\Lambda(z)}{70}\right) \right]^{-1} \quad (\text{B.21})$$

$$\Omega(z) = \Omega_0(1+z)^3 g^{-2}(z), \quad \Omega_\Lambda(z) = \Omega_\Lambda g^{-2}(z) \quad (\text{B.22})$$

$$g^2(z) = (1 - \Omega_0 - \Omega_\Lambda)(1+z)^2 + \Omega_0(1+z)^3 + \Omega_\Lambda \quad (\text{B.23})$$

$$T_{sup}(k) = \frac{L}{L + C q_{\text{eff}}^2} \quad (\text{B.24})$$

$$L = \ln(e + 1.84\beta_c\sqrt{\alpha_\nu}q_{\text{eff}}), \quad C = 14.4 + \frac{325}{1 + 60.5q_{\text{eff}}^{1.08}}, \quad \beta_c = (1 - 0.949f_{\nu b})^{-1} \quad (\text{B.25})$$

$$y_{fs}(q, f_\nu) = 17.2f_\nu(1 + 0.488f_\nu^{-7/6}) \left(\frac{qN_\nu}{f_\nu}\right)^2, \quad q = \frac{k}{\text{Mpc}^{-1}}\Theta_{2.7}^2\omega_0^{-1} \quad (\text{B.26})$$

$$B(k) = 1 + \frac{1.24f_\nu^{0.64}N_\nu^{0.3+0.6f_\nu}}{q_\nu^{-1.6} + q_\nu^{0.8}}, \quad q_\nu = 3.92q\sqrt{\frac{N_\nu}{f_\nu}} \quad (\text{B.27})$$

$$f_b = \frac{\Omega_b}{\Omega_m + \Omega_\nu}, \quad f_\nu = \frac{\Omega_\nu}{\Omega_m + \Omega_\nu}, \quad f_{cb} = \frac{\Omega_m}{\Omega_m + \Omega_\nu}, \quad f_{\nu b} = \frac{\Omega_b + \Omega_{nu}}{\Omega_m + \Omega_\nu} \quad (\text{B.28})$$

$$q_{\text{eff}} = \frac{k\Theta_{2.7}^2}{\Gamma_{\text{eff}} \text{Mpc}^{-1}}, \quad \Gamma_{\text{eff}} = \omega_0 \left[\sqrt{\alpha_\nu} + \frac{1 - \sqrt{\alpha_\nu}}{1 + (0.43ks)^4} \right] \quad (\text{B.29})$$

$$\begin{aligned} \alpha_\nu = & \frac{f_c}{f_{cb}} \frac{5 - 2(p_c + p_{cb})}{5 - 4p_{cb}} \frac{1 - 0.553f_{\nu b} + 0.126f_{\nu b}^3}{1 - 0.193\sqrt{f_\nu N_\nu} + 0.169f_\nu N_\nu^{0.2}} (1 + y_d)^{p_{cb} - p_c} \\ & \times \left[1 + \frac{p_c - p_{cb}}{2} \left(1 + \frac{1}{(3 - 4p_c)(7 - 4p_{cb})} \right) (1 + y_d)^{-1} \right] \end{aligned} \quad (\text{B.30})$$

$$p_c = \frac{1}{4} \left[5 - \sqrt{1 + 24f_c} \right], \quad p_{cb} = \frac{1}{4} \left[5 - \sqrt{1 + 24f_{cb}} \right] \quad (\text{B.31})$$

$$s = \frac{44.5 \ln(9.83\omega_0)}{\sqrt{1 + 10\omega_b^{3/4}}} \text{ Mpc} \quad (\text{B.32})$$

Where the definitions of z_{eq} , z_d and y are the same as in section B.1. In addition we have to note that N_ν is the number of massive neutrinos and we have to redefine $\Omega_0 = \Omega_c + \Omega_b + \Omega_\nu$, and so $\omega_0 = \Omega_0 h^2$. Also in this case, we recommend reading [43] for the meaning of all these parameters.

Bibliography

- [1] R. M. Wald, *General relativity*. University of Chicago press, 2010.
- [2] L. P. Eisenhart, *Riemannian geometry*. Princeton university press, 1997, vol. 51.
- [3] S. Akcay and R. A. Matzner, “The kerr-de sitter universe,” *Classical and Quantum Gravity*, vol. 28, no. 8, p. 085012, 2011.
- [4] M. S. Longair, *Galaxy formation*. Springer Science & Business Media, 2007.
- [5] P. Hibbs. Rotation curve of a typical spiral galaxy. Available: <https://commons.wikimedia.org/wiki/File:GalacticRotation2.svg>
- [6] NASA. Timeline of the universe. Available: <https://map.gsfc.nasa.gov/media/060915/index.html>
- [7] N. Aghanim *et al.*, “Planck 2018 results-vi. cosmological parameters,” *Astronomy & Astrophysics*, vol. 641, p. A6, 2020.
- [8] G. F. Smoot, “Cobe observations and results,” in *AIP Conference Proceedings CONF-981098*, vol. 476, no. 1. American Institute of Physics, 1999, pp. 1–10.
- [9] A. G. Riess, S. Casertano, W. Yuan, L. M. Macri, and D. Scolnic, “Large magellanic cloud cepheid standards provide a 1% foundation for the determination of the hubble constant and stronger evidence for physics beyond Λ cdm,” *The Astrophysical Journal*, vol. 876, no. 1, p. 85, 2019.
- [10] E. Abdalla *et al.*, “Cosmology intertwined: A review of the particle physics, astrophysics, and cosmology associated with the cosmological tensions and anomalies,” *Journal of High Energy Astrophysics*, vol. 34, pp. 49–211, 2022.
- [11] D. Tong, *Lecture on cosmology*. University of Cambridge, 2009.
- [12] ESA. The anisotropies of the cmb as observed by planck. Available: https://www.esa.int/ESA_Multimedia/Images/2013/03/Planck_CMB
- [13] S. Dodelson and F. Schmidt, *Modern cosmology*. Academic press, 2020.
- [14] H. Winther. Cmb power spectra. Available: <https://cmb.wintherscoming.no/milestone4.php>

- [15] P. Callin, “How to calculate the cmb spectrum,” 2006.
- [16] J. M. Bardeen, J. Bond, N. Kaiser, and A. Szalay, “The statistics of peaks of gaussian random fields,” *Astrophysical Journal, Part 1 (ISSN 0004-637X)*, vol. 304, May 1, 1986, p. 15-61. SERC-supported research., vol. 304, pp. 15–61, 1986.
- [17] M. S. Turner, “The road to precision cosmology,” *Annual Review of Nuclear and Particle Science*, vol. 72, no. 1, pp. 1–35, sep 2022.
- [18] D. J. Eisenstein and W. Hu, “Baryonic features in the matter transfer function,” *The Astrophysical Journal*, vol. 496, no. 2, p. 605, 1998.
- [19] H. Winther. Cosmological perturbation theory. Available: https://cmb-wintherscoming.no/theory_perturbations.php
- [20] E. Bellini *et al.*, “Comparison of einstein-boltzmann solvers for testing general relativity,” *Physical Review D*, vol. 97, no. 2, p. 023520, 2018.
- [21] A. Lewis, A. Challinor, and A. Lasenby, “Efficient computation of cosmic microwave background anisotropies in closed friedmann-robertson-walker models,” *The Astrophysical Journal*, vol. 538, no. 2, p. 473, 2000.
- [22] J. Lesgourgues, “The cosmic linear anisotropy solving system (class) i: overview,” *arXiv preprint arXiv:1104.2932*, 2011.
- [23] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.
- [24] IBM. What are neural networks? Available: <https://www.ibm.com/topics/neural-networks>
- [25] F. Marquardt, “Machine learning and quantum devices,” *SciPost Physics Lecture Notes*, p. 029, 2021.
- [26] G. Cybenko, “Approximation by superpositions of a sigmoidal function,” *Mathematics of control, signals and systems*, vol. 2, no. 4, pp. 303–314, 1989.
- [27] I. A. Abdellaoui and S. Mehrkanoon, “Symbolic regression for scientific discovery: an application to wind speed forecasting,” in *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2021, pp. 01–08.
- [28] M. Cranmer, “Interpretable machine learning for science with pysr and symbolicregression. jl,” *arXiv preprint arXiv:2305.01582*, 2023.
- [29] P. Gasper, K. Gering, E. Dufek, and K. Smith, “Challenging practices of algebraic battery life models through statistical validation and model identification via machine-learning,” *Journal of the Electrochemical Society*, vol. 168, no. 2, p. 020502, 2021.

- [30] S. Stijven, E. Vladislavleva, A. Kordon, L. Willem, and M. E. Kotanchek, “Prime-time: Symbolic regression takes its place in the real world,” *Genetic Programming Theory and Practice XIII*, pp. 241–260, 2016.
- [31] P. Lemos, N. Jeffrey, M. Cranmer, S. Ho, and P. Battaglia, “Rediscovering orbital mechanics with machine learning,” *arXiv preprint arXiv:2202.02306*, 2022.
- [32] J. B. Orjuela-Quintana, S. Nesseris, and W. Cardona, “Using machine learning to compress the matter transfer function $t(k)$,” *Physical Review D*, vol. 107, no. 8, p. 083520, 2023.
- [33] M. Mitchell, “Genetic algorithms: An overview.” in *Complex.*, vol. 1, no. 1. Citeseer, 1995, pp. 31–39.
- [34] C. G. Broyden, “The convergence of a class of double-rank minimization algorithms 1. general considerations,” *IMA Journal of Applied Mathematics*, vol. 6, no. 1, pp. 76–90, 1970.
- [35] D. R. Stump, “Precession of the perihelion of mercury,” *American Journal of Physics*, vol. 56, no. 12, pp. 1097–1098, 1988.
- [36] S. Nadkarni-Ghosh and A. Refregier, “The einstein–boltzmann equations revisited,” *Monthly Notices of the Royal Astronomical Society*, vol. 471, no. 2, pp. 2391–2430, 2017.
- [37] A. G. Sánchez, C. M. Baugh, and R. Angulo, “What is the best way to measure baryonic acoustic oscillations?” *Monthly Notices of the Royal Astronomical Society*, vol. 390, no. 4, pp. 1470–1490, 2008.
- [38] K. Abe *et al.*, “Solar neutrino results in super-kamiokande-iii,” *Physical Review D*, vol. 83, no. 5, p. 052010, 2011.
- [39] J. Sakurai, “Rest mass of the neutrino,” *Physical Review Letters*, vol. 1, no. 1, p. 40, 1958.
- [40] M. Cranmer. PySRegressor reference. Available: <https://astroautomata.com/PySR/api/>
- [41] M. Bonici, F. Bianchini, and J. Ruiz-Zapatero, “Capse. jl: efficient and auto-differentiable cmb power spectra emulation,” *arXiv preprint arXiv:2307.14339*, 2023.
- [42] L. Pagano, J.-M. Delouis, S. Mottet, J.-L. Puget, and L. Vibert, “Reionization optical depth determination from planck hfi data with ten percent accuracy,” *Astronomy & Astrophysics*, vol. 635, p. A99, 2020.
- [43] D. J. Eisenstein and W. Hu, “Power spectra for cold dark matter and its variants,” *The Astrophysical Journal*, vol. 511, no. 1, p. 5, 1999.