INSTITUTO POLITÉCNICO NACIONAL

Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada - Legaria

# Artificial neural networks in Bayesian inference

TESIS

QUE PARA OBTENER EL GRADO DE

DOCTOR EN TECNOLOGÍA AVANZADA

PRESENTA

**Isidro Gómez Vargas**

Directores de Tesis:

Dr. Ricardo García Salcedo
CICATA-Legaria del Instituto Politécnico Nacional
y
Dr. José Alberto Vázquez González
Instituto de Ciencias Físicas de la Universidad Nacional Autónoma de México

Ciudad de México Febrero 2021

SIP-13
REP 2017

# INSTITUTO POLITÉCNICO NACIONAL

## SECRETARIA DE INVESTIGACIÓN Y POSGRADO

*ACTA DE REGISTRO DE TEMA DE TESIS*
*Y DESIGNACIÓN DE DIRECTOR DE TESIS*

Ciudad de México 27 de octubre del 2020

El Colegio de Profesores de Posgrado de Tecnología Avanzada en su Sesión
(Unidad Académica)

Reunión ordinaria No XI celebrada el día 27 del mes octubre de 2020 , conoció la solicitud presentada por el alumno:

| | | | | | |
|---|---|---|---|---|---|
| **Apellido Paterno:** | **Gómez** | **Apellido Materno:** | **Vargas** | **Nombre (s):** | **Isidro** |

Número de registro: A 1 7 0 8 5 1

del Programa Académico de Posgrado: Doctorado en Tecnología Avanzada

Referente al registro de su tema de tesis; acordando lo siguiente:

1.- Se designa al aspirante el tema de tesis titulado:

Artificial neural networks in Bayesian inference

Objetivo general del trabajo de tesis:

Implementar algoritmos estadísticos para la estimación de parámetros, comparación de modelos teóricos y análisis de datos observacionales, en particular, de cosmología. Emplear redes neuronales artificiales para optimizar procesos de inferencia bayesiana y hacer reconstrucciones de funciones cosmológicas a partir de los datos.

2.- Se designa como Directores de Tesis a los profesores:

Director: Ricardo García Salcedo 2° Director: José Alberto Vázquez González

No aplica: □

3.- El Trabajo de investigación base para el desarrollo de la tesis será elaborado por el alumno en:

Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada unidad Legaria del Instituto Politécnico Nacional

que cuenta con los recursos e infraestructura necesarios.

4.- El interesado deberá asistir a los seminarios desarrollados en el área de adscripción del trabajo desde la fecha en que se suscribe la presente, hasta la aprobación de la versión completa de la tesis por parte de la Comisión Revisora correspondiente.

Director de Tesis Ricardo García

2° Director de Tesis

Aspirante

Presidente del Colegio

ESTADOS UNIDOS MEXICANOS

SECRETARÍA
DE
EDUCACIÓN PÚBLICA

# INSTITUTO POLITÉCNICO NACIONAL

SIP-17-CI

## SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

## *ACTA DE EXAMEN DE GRADO DE DOCTORADO*

En la Ciudad de **México**, a las **10:00** horas del día **19** del mes de **febrero** del año **2021** reunidos en el **Aula Magna** designado para tal efecto, los Profesores del **Centro**:
**Dr. Teodoro Rivera Montalvo, Dr. Antonio Gustavo Juárez Gracia, Dr. Ricardo García Salcedo,**
**Dr. José Alberto Vázquez González, Dr. Israel Quiros Rodríguez**

designados para integrar el Jurado de Examen de Grado de: **DOCTORADO EN TECNOLOGÍA AVANZADA**

de: **Isidro Gómez Vargas**

Con registro: **A170851** y considerando que ha cumplido con los requisitos correspondientes, se procedió a efectuar el examen en los términos que establece el Reglamento de Estudios de Posgrado. Después de concluir la disertación y réplica de rigor, el jurado deliberó, habiéndose obtenido el siguiente resultado:

**Aprobado**

Para constancia se levantó la presente acta a las **13:00** horas del día **19** del mes de **febrero** del año **2021**, misma que Suscriben los sinodales mencionados.

**PRESIDENTE**

**Dr. Teodoro Rivera Montalvo**

**SECRETARIO**

**Dr. Antonio Gustavo Juárez Gracia**

**1er VOCAL**

Ricardo García

**Dr. Ricardo García Salcedo**

**2º VOCAL**

**Dr. José Alberto Vázquez González**

**TESIS**

*"Artificial neural networks in Bayesian inference"*

**3er VOCAL**

**Dr. Israel Quirós Rodríguez**

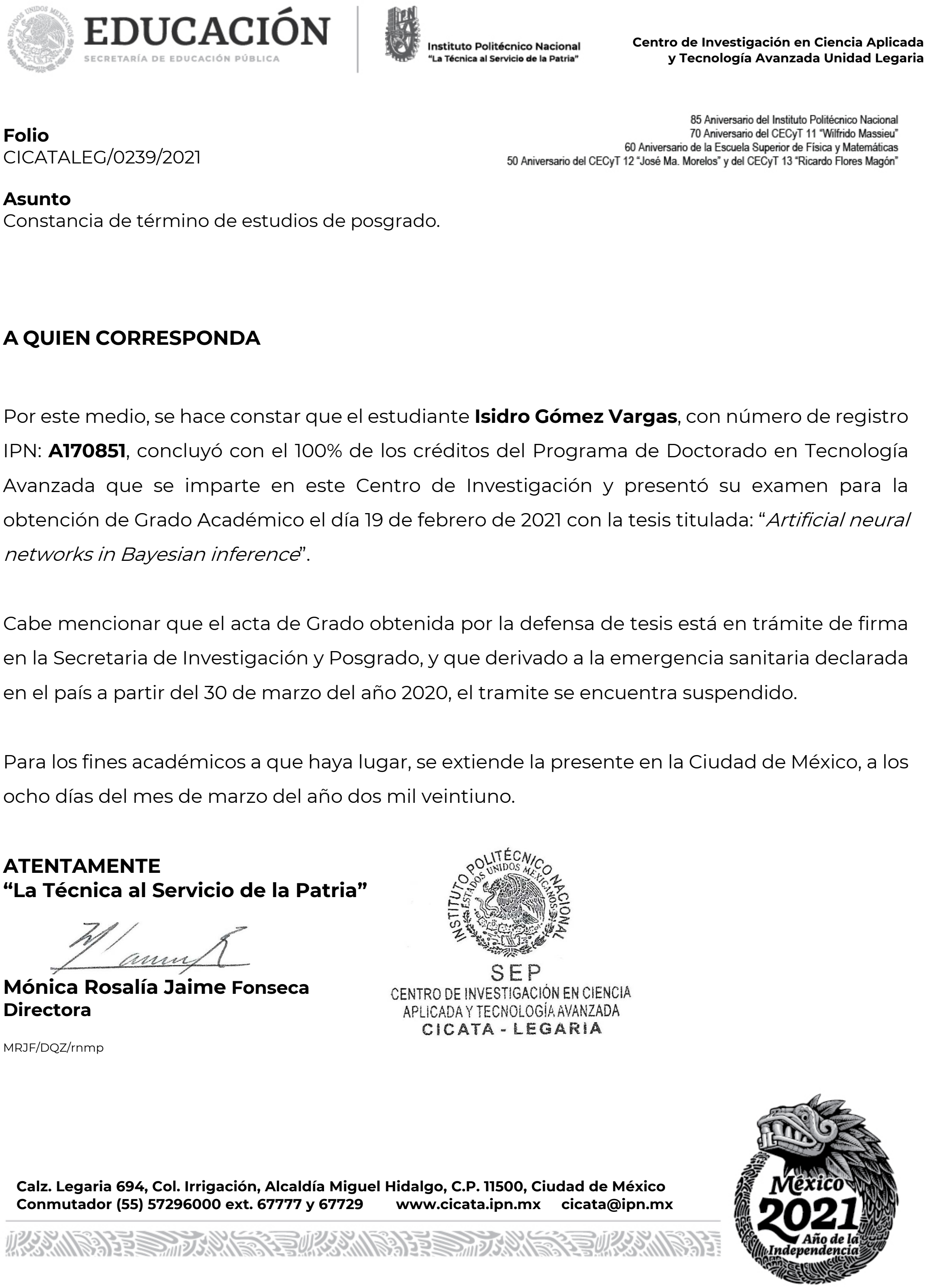
**Secretario de Investigación y Posgrado**

**Dr. Juan Silvestre Aranda Barradas**

**EL SUSCRITO DIRECTOR DEL Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada-Legaria**
**CERTIFICA** que las firmas que anteceden son auténticas y corresponden a las personas cuyos nombres aparecen en esta acta.

Dra. Mónica Rosalía Jaime Fonseca

INSTITUTO POLITÉCNICO NACIONAL
ESTADOS UNIDOS MEXICANOS
SEP
CENTRO DE INVESTIGACIÓN EN CIENCIA
APLICADA Y TECNOLOGÍA AVANZADA
CICATA - LEGARIA



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

Instituto Politécnico Nacional
"La Técnica al Servicio de la Patria"

Centro de Investigación en Ciencia Aplicada
y Tecnología Avanzada Unidad Legaria

85 Aniversario del Instituto Politécnico Nacional
70 Aniversario del CECyT 11 "Wilfrido Massieu"
60 Aniversario de la Escuela Superior de Física y Matemáticas
50 Aniversario del CECyT 12 "José Ma. Morelos" y del CECyT 13 "Ricardo Flores Magón"

**Folio**
CICATALEG/0239/2021

**Asunto**
Constancia de término de estudios de posgrado.

**A QUIEN CORRESPONDA**

Por este medio, se hace constar que el estudiante **Isidro Gómez Vargas**, con número de registro IPN: **A170851**, concluyó con el 100% de los créditos del Programa de Doctorado en Tecnología Avanzada que se imparte en este Centro de Investigación y presentó su examen para la obtención de Grado Académico el día 19 de febrero de 2021 con la tesis titulada: "*Artificial neural networks in Bayesian inference*".

Cabe mencionar que el acta de Grado obtenida por la defensa de tesis está en trámite de firma en la Secretaria de Investigación y Posgrado, y que derivado a la emergencia sanitaria declarada en el país a partir del 30 de marzo del año 2020, el tramite se encuentra suspendido.

Para los fines académicos a que haya lugar, se extiende la presente en la Ciudad de México, a los ocho días del mes de marzo del año dos mil veintiuno.

**ATENTAMENTE**
**"La Técnica al Servicio de la Patria"**

**Mónica Rosalía Jaime Fonseca**
**Directora**

INSTITUTO POLITÉCNICO NACIONAL
ESTADOS UNIDOS MEXICANOS
SEP
CENTRO DE INVESTIGACIÓN EN CIENCIA
APLICADA Y TECNOLOGÍA AVANZADA
CICATA - LEGARIA

MRJF/DQZ/rnmp

Calz. Legaria 694, Col. Irrigación, Alcaldía Miguel Hidalgo, C.P. 11500, Ciudad de México
Conmutador (55) 57296000 ext. 67777 y 67729 www.cicata.ipn.mx cicata@ipn.mx

México
2021
Año de la
Independencia

# *INSTITUTO POLITÉCNICO NACIONAL*

## *SECRETARÍA DE INVESTIGACIÓN Y POSGRADO*

## *CARTA CESIÓN DE DERECHOS*

En la Ciudad de México el día 20 del mes Febrero del año 2021, el que suscribe Isidro Gómez Vargas alumno (a) del Programa de Doctorado en Tecnología Avanzada con número de registro A170851, adscrito a el Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada Unidad Legaria CICATA-Legaria, manifiesta que es autor (a) intelectual del presente trabajo de Tesis bajo la dirección del Dr. Ricardo García Salcedo y del Dr. José Alberto Vázquez González y cede los derechos del trabajo intitulado Artificial neural networks in Bayesian inference, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección igomezv0701@alumno.ipn.mx. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

Isidro Gómez Vargas

# Declaration

In Mexico city, on 20 February 2021, the undersigned Isidro Gómez Vargas, student of the Advanced Technology PhD program with registration number A170851, affiliated to the Centre for Research in Applied Science and Advanced Technology (CICATA-Legaria), states that he is the intellectual author of this thesis work, under the supervision of Dr. Ricardo García Salcedo and Dr. José Alberto Vázquez González, and transfers the rights of the work entitled *Artificial Neural Networks in Bayesian Inference* to the National Polytechnic Institute for outreach, academic and research purposes.

The work in this PhD thesis is based on research carried out at the CICATA-Legaria of the National Polytechnic Institute in collaboration with the Centre of Physical Sciences of the National Autonomous University of Mexico. Portions of this thesis has been published in some papers and conference proceedings.

## Publications:

- Gómez-Vargas, I., Medel-Esquivel, R., García-Salcedo, R. & Vázquez, JA (2021). *Cosmological reconstructions with artificial neural networks*. Subbmited to Journal of Cosmology and Astrophysics. IOP Publishing. [Arxiv:2104.00595].
- Gómez-Vargas, I., Medel-Esquivel, R., García-Salcedo, R. & Vázquez, JA (2021). *Neural networks within a Bayesian inference framework*. Journal of Physics: Conference Series. (Vol. 1723, No. 1, p. 012022). IOP Publishing.
- Medel Esquivel, R., Gómez-Vargas, I., Vázquez, J. A., & García-Salcedo, R. (2021). *An introduction to Markov Chain Monte Carlo*. Boletín de Estadística e Investigación Operativa. (Vol. 37, No. 1, p. 47-74). Sociedad de Estadística e Investigación Operativa.
- Vázquez J. A., Medel Esquivel, R & Gómez-Vargas, I. (2021) Cosmología observacional con Redes Neuronales Artificiales. Memorias de la XXVII Escuela de Verano en Física (Vol. 27, No. 1, p. 89). Instituto de Física e Instituto de Ciencias Físicas de la UNAM. ISSN: 2594-2697.
- Gómez-Vargas, I., Medel Esquivel, R., García Salcedo, R. & Vázquez JA (2019). *Una Aplicación de las Redes Neuronales Artificiales en la Cosmología*. Komputer Sapiens (Vol. 2, No. 11, p. 12-17). Sociedad Mexicana de Inteligencia Artificial.

# RESUMEN

Las Redes Neuronales Artificiales son modelos computacionales con la capacidad de aproximar cualquier función no lineal, lo que permite incorporarlas en el modelado de datos y en su análisis estadístico. El principal objetivo de esta tesis ha sido mostrar la pertinencia del uso de las redes neuronales dentro del análisis Bayesiano de datos mediante dos diferentes maneras: 1) reduciendo el tiempo de la inferencia Bayesiana y 2) creando modelos para los datos para después analizar datos nuevos (generados por estos modelos de redes neuronales) mediante inferencia Bayesiana. Hemos utilizado conjuntos de datos cosmológicos para validar nuestros métodos y, por lo tanto, también hemos demostrado que se puede extraer información cosmológica interesante a partir de estos enfoques, en particular, los modelos de redes neuronales pueden evidenciar ciertas problemáticas que actualmente presenta el modelo estándar de la cosmología ΛCDM utilizando, solamente, conjuntos de datos pequeños de observaciones cosmológicas con corrimientos al rojo menores que $z = 2$.

**Palabras clave:** *Inferencia bayesiana, redes neuronales artificiales, cosmología observacional*

# ABSTRACT

Artificial Neural Networks are computational models with the ability to approximate any non-linear function, which allows them to be incorporated into data modelling and statistical analysis. The main goal of this thesis has been to show the relevance of the use of neural networks within Bayesian data analysis by two different ways: 1) reducing the time of Bayesian inference and 2) creating models for the data and then analysing the new data (generated by these neural network models) through Bayesian inference. We have used cosmological datasets to validate our methods, and therefore we have also shown that interesting cosmological information can be extracted from these approaches, in particular, neural network models can highlight certain drawbacks currently presented by the standard model of cosmology $\Lambda$CDM using only small datasets of cosmological observations with redshifts less than $z = 2$.

**Keywords:** *Bayesian inference, Artificial neural networks, observational cosmology*

# AGRADECIMIENTOS

Thanks to my family, advisors and friends.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# CHAPTER 1

## OUTLINE

Modelling is a cornerstone in science. Scientists need to create models to explain a nature phenomena and improve the knowledge about it. A deductive way to generate a model is only based on theoretical principles and physical laws through the formulation of equations. Another approach to create a model is using the experimental (or measured) data and, inductively, to infer the relationships between several variables. This last approach based on *learning from the data* is the spirit of data analysis and in this thesis we have focused on two approaches: Bayesian inference and non-parametric modelling.

In Bayesian modelling there are assumed some probabilistic density functions to the relationships of the parameters of a theoretical model and the data through the Bayes' theorem and using sampling algorithms the obtained output is a set of samples that can be described by a probabilistic density function that corresponds to the conditional probability of the parameters of the model given the data. The samples allow to know the median and standard deviations of their, beforehand unknown, probability distribution.

In recent years, the increase in hardware power and the maturity of various computational techniques for storing, analyzing and processing data have allowed the consolidation of the called *Data Science*, in which the *Machine Learning* is the part of our interest in this thesis because it is the field of the Artificial Intelligence dedicated to the statistical modelling for the data, in this sense, Machine Learning is the heir of the traditional statistics in the current computational kingdom.

Artificial Neural Networks, computational models of Machine Learning, have shone by their own light and their great applicability in various scientific, medical, industrial and social sectors, and therefore, a new field of study has been inaugurated dedicated to them: *Deep Learning*. In the present work, we have applied Artificial Neural Networks in two different tasks: 1) to reduce the computational time of a Bayesian inference process by a nested sampling algorithm and 2) to perform non-parametric reconstructions of the intrinsic functions in the data.

All the data used in this work is from cosmological observations and therefore some of the analysis have cosmological implications. We choose this nature of data because we know well the theory, the cosmological models have several parameters and Bayesian inference is widely used in this scientific field. However the type of analysis and methods shown in this work can be implemented to any other type of datasets.

The task of non-parametric reconstructions of cosmological data falls into the non-parametric statistical inference because the process made with the Artificial Neural Networks, over the cosmological data, does not have any theoretical or statistical assumption beforehand and allows new values predictions based on the existing data. Once the Artificial Neural Networks have been properly trained, a computational model for the data have been produced and it have the faculty of generate new data with which a Bayesian inference process, based on a theoretical model, can be performed to analyze the insights of these new datasets and make a comparison with the originals.

The aim of this thesis is to show some strategies to include the modelling with Artificial Neural Networks into the traditional Bayesian analysis, in particular (but not exclusively) in the field of observational cosmology. We show that use of Artificial Neural Network can reduce the time and complement the quality of a cosmological data analysis based on Bayesian inference.

## 1.1 Chapter overview

In the first chapters of this thesis, we provide the worth theoretical frameworks: statistical, Artificial Neural Networks and cosmology. In the Section 2 we show the basis of Bayesian inference and the Markov Chain Monte Carlo algorithms; also we describe the general idea of the non-parametric approach. The Section 2.4 have an overview of the basis of Artificial Neural Networks and, in particular, about the Feed Forward Neural Networks, convolutional neural networks and Autoencoders. The Chapter 3 contains the worth cosmological background to understand the cosmological datasets used in this work, this chapter also describes the cosmological parameter estimation code *SimpleMC* and my contributions on it.

The chapters 4, 5 and 6 have the applications of Artificial Neural Networks in the analysis of cosmological data. Chapter 4 shows a method to reduce the computational time of a Bayesian inference process learning the likelihood function on real time [2–4], Chapter 5 have a non-parametric approach to modeling several cosmological data and the Chapter 6 is and extension of the previous chapter and contains the reconstructions of covariance matrix with a variational autoencoder [5]. Finally, Chapter 7 shows the conclusions and summary about all this manuscript.

# CHAPTER 2

## STATISTICAL BACKGROUND

### 2.1 Introduction

The strength of the probability theory, therefore also of statistics, is well synthesized in a phrase of Pierre-Simon Laplace of his *Philosophical Essay on Probabilities*:

> "...the theory of probabilities is basically only common sense reduced to a calculus..."

Based on the above, what better than to use statistics as the mathematical tool par excellence to analyze data. In all scientific areas, we often have models and data, therefore we need to make inferences. The word inference refers at the process to obtain logical consequences assuming some premises. Statistical inference, also called, inductive statistics, is a way of reasoning from sample data to population parameters, for example, any prediction, generalization, prediction, decision or estimation based on a sample data [6]. In this sense, data analysis, machine learning and data mining are different names to the practice of statistical inference according diverse contexts [7]. Indeed, the most popular task of machine learning and data mining such as clustering, classification or prediction are different applications of statistical inference [7, 8].

In the computer science vocabulary, statistical inference is known as learning. Its goal is to obtain some idea about the distribution that obey a given dataset and when a computational model or algorithm discovers some pattern in the insights of the data, then it had learned. There are two types of approaches to analyze the data: parametric or non-parametric.

To make possible a parametric inference it is necessary to have a probabilistic model with a finite number of parameters. On the other hand, the non-parametric inference subtracts information directly to the data with the less possible assumptions or with models that have an undetermined number of parameters.

In this work we used, Bayesian inference as parametric approach. In particular, we use Markov Chain Monte Carlo (MCMC) methods. In addition, we implement Artificial Neural Networks to develop non-parametric inferences.

In this chapter, we show an overview about statistical inference, MCMC methods, a briefly idea about non-parametric inference and the necessary background about Artificial Neural Networks. We do not make a deep description of classical non-parametric techniques because we do not use them in the present work.

## 2.2 Bayesian inference

Bayesian inference is a paradigm to infer unknown quantities of a theoretical model using experimental data based on previous knowledge or assumptions, for this reason it is also referred as a subjective thinking; however it is a very robust mathematical method that along the years has been tested in several fields. Bayesian inference works with the Bayes' theorem that involves conditional probabilities.

The classical derivation of the Bayes' theorem, using the basic rules of sum and product of probabilities, have the following expression:

$$P(A_j|E) = \frac{P(E|A_j)P(A_j)}{\sum_{i=1}^{k} P(A_i)P(E|A_i)}, \quad j = 1,2,...k \tag{2.1}$$

where $A_1, A_2, ..., A_k$ are $k$ independent events and each probability is based on known events and their frequencies.

The use of the Bayes' theorem does not necessary falls under the *Bayesian* paradigm. The Bayesian perspective does not use frequentist probabilities and use the theorem to measures the uncertainty of the data, below in this section we talk more about the called Bayesian methods.

We after mentioned that to perform Bayesian inference it is necessary a dataset. This dataset can be from experiments, observations or other source; furthermore, this data should to have the relevant information about the phenomena under study. Thus we can assume that a dataset of measurements $D$ under a non-linear model have the following general form:

$$D = f(x;\theta) + \varepsilon, \tag{2.2}$$

where $x$ represents the known quantities such as control variables or constants; $\theta = (\theta_1, \theta_2, ..., \theta_N)$ is the vector of unknown parameters and $\varepsilon$ indicates the measurement errors. With this nomenclature, we can write the Bayes' theorem, assuming some *believes* about the probabilities, and convert all the terms involved into probability density functions (PDFs):

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)}, \tag{2.3}$$

where $P(\theta)$ is the prior distributions over the parameters $\theta$ and can represent previous knowledge about the parameters before the data are observed, if we do not know anything about the involved parameters, the prior PDF can be an *uninformative* prior $P(\theta) = 1$; if we know the bounds of the parameters we can define the prior as an uniform distribution; another common practice is to choose a conjugate prior, that means the posterior PDF and prior PDF are of the same family of distributions, for example, exponential and normal distributions. On the other hand, $P(D|\theta)$ is the likelihood function and indicates the conditional probability of the data given the model. $P(D)$ is a normalisation constant, that is, the likelihood marginalisation and is called Bayesian evidence.

$$P(D) = \int_{\mathbb{R}^N} P(D|\theta)P(\theta)d\theta, \tag{2.4}$$

where $N$ is the number of dimensions of the parameter space for $\theta$. When there are more than a few dimensions, this integral is very hard to estimate, often impossible analytically, and turns very difficult the calculation of the posterior PDF through the Bayes' theorem. The sampling methods of Markov Chain Monte Carlo (see Section 2.2.4) avoid this problem using rates of posteriors in order to cancel this normalisation constant. On the other hand, algorithms such as nested sampling (Section 2.2.5) allow to calculate this quantity during the sampling.

It can be assumed that the measurement error $\varepsilon$ is independent of $\theta$ and has a PDF $P_\varepsilon$. In this case, the predicted value and the measurement error share the same distribution, therefore the likelihood function can be expressed as:

$$P(D|\theta) = P_\varepsilon(D - f(x;\theta)), \tag{2.5}$$

and if the error $\varepsilon \sim N(0,C)$ has a normal distribution centered in zero and a covariance matrix C, then:

$$P(D|\theta) = \frac{1}{(2\pi)^{n/2}|C|^{1/2}} e^{-0.5(D-f(x;\theta)^T C^{-1}(D-f(x;\theta))} \ , \tag{2.6}$$

Bayesian inference can perform two important tasks in data analysis: parameter estimation and model comparison.

### 2.2.1 Parameter estimation

One of the goals of Bayesian inference is to know the values and uncertainties of the most probable $\theta$ parameters of a mathematical model. This task is called parameter estimation.

The problem to find the value of $\theta$ most probable under the likelihood PDF is known as Maximum Likelihood Estimation (MLE): $\theta_{MLE} = max_\theta P(D|\theta)$. In the Bayesian context, if the prior PDF is uninformative, the find of this value is analog to find the most probable value of the parameter vector in the posterior PDF (Maximum A Posteriori, MAP). However, for informative priors, we need to find the maximum value of the parameter vector to the posterior , MAP) considering both the likelihood and the prior PDFs:

$$\theta_{MAP} = max_\theta(P(D|\theta)P(\theta)); \tag{2.7}$$

In order to find MAP or MLE, there are several techniques such as optimization (eg. simplex, gradient descent, Newton), meta-heuristics (eg. genetic algorithms, particle swarm optimization), approximation (eg. Laplace and variational), quadrature, Monte Carlo and Markov Chain Monte Carlo (see [9] for more details). We will focus on MCMC methods (Section 2.2.4), but before it is worth a very briefly introduction abut what is a Monte Carlo method included in Section 2.2.3.

### Example: Fitting a straight line

Suppose we have a dataset and a linear model is proposed to describe it $y = ax + b$. The parameters of the model are the abscissa $a$ and the ordinate to the origin $b$. By performing Bayesian inference for this model and with the data (black dots in the left panel of Figure 2.1) we find the most probable straight line to describe the data (red line). In addition, Bayesian inference provides us their uncertainties with the posterior PDFs of these parameters, for example, in the right panel of Figure 2.1 the 1D plots show the posterior probability density function of each parameter separately, while the 2D plot is the joint PDF, where the darker region of the ellipse indicates the region corresponding to the $1\sigma$ deviation and the less dark region to $2\sigma$.

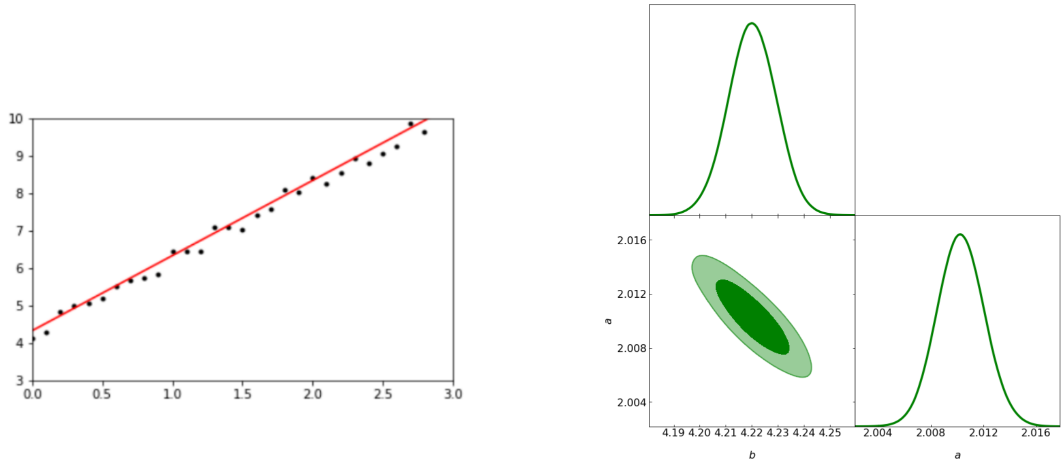


Figure 2.1: Example parameter estimation of a straight line

## 2.2.2 Model Comparison

In each dataset under analysis, different mathematical models may be proposed to describe them, however, we know that if there is a correct description of the data, only one model could be true. Therefore, the comparison of models is very important.

There are some measures called information criteria [10, 11] among which are the Akaike Information Criteria (AIC) and Bayesian Information Criteria (BIC), for example.

However, these measures are very sensitive to the size of the data set and the number of parameters of the model under analysis, so they do not provide completely reliable information in many circumstances. On the other hand, Bayesian evidence $Z$ has been shown to be a much more robust and rigorous mathematical tool [1, 12] , which by means of the Bayes Factor allows the comparison of models. The Bayes factor $B_{0,1}$ of the Model 0 with respect to Model 1 is the ratio of their respective Bayesian evidences:

$$B_{0,1} = \frac{Z_0}{Z_1} \tag{2.8}$$

or in logarithm:

$$\ln B_{0,1} = \ln Z_0 - \ln Z_1 \tag{2.9}$$

For a more detailed discussion on the advantage of comparing models with the Bayes factor over information criteria, see reference [13].

In order to comparing two models through Bayesian factors, it can be used the scale proposed by Harold Jeffrey shown in Table 2.1 [14]. In the Table 2.1 the strength of the Bayesian evidence $Z$ is in favours of the Model 0 over the Model 1.

| $lnB_{0,1}$ | Strength of $Z$ |
|---|---|
| $< 1$ | Inconclusive |
| $1 - 2.5$ | Significant |
| $2.5 - 5$ | Strong |
| $> 5$ | Decisive |

Table 2.1: Jeffreys' scale for comparison between two models.

.

## Example: Toy models comparison

We generate a synthetic data set with the function $y = -3.5x^2 + 3.6x - 0.1$ plus random noise. Therefore, we can assume that this is our experimental data set and propose the following models to describe the data:

- Model 1: $y = ax^2 + (-a+b)x + c$.

- Model 2: $y = a\sin(bx) + c$

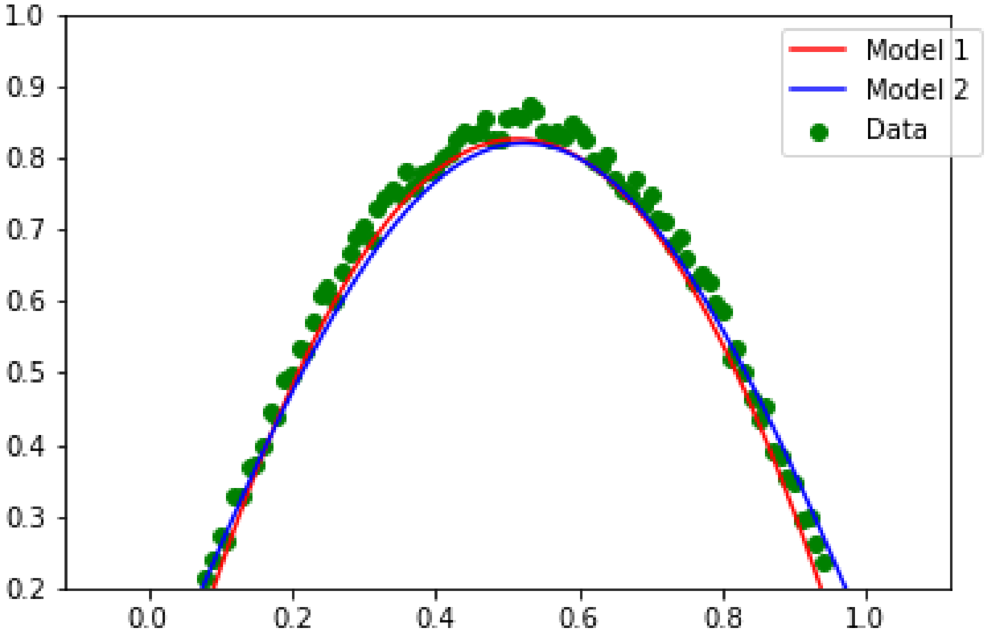


Figure 2.2: Example: Model comparison

Figure 2.2 shows that these two models, at first glance, are very similar to the data. However, let us perform the calculation of the Bayes factor for these models:

$$\begin{aligned} B_{1,2} =& log(\frac{Z_1}{Z_2}) \\ &= log(Z_1) - log(Z_2) \\ &= -62.002 - (-255.064) \\ &= 193.062 \end{aligned}$$

The Bayes factor $B_{1,2} = 193.062$ indicates, according to Jeffrey's scale, that there is a decisive advantage for model 1, which was to be expected due to the way in which the dataset was generated and, therefore, the power of model comparison by Bayesian evidence can be appreciated in this example.

### 2.2.3 Monte Carlo methods

To understand the spirit of Monte Carlo methods, it is worth to remember some probability laws. For more details, we recommend the didactic paper of the reference [15] or its code repository [1].

**Theorem 2.2.1** ***The Weak Law of Large Numbers***

*If $X_1, X_2, ... X_n$ are independent and identically distributed (iid) with mean $\mu$. Then, for each $\varepsilon > 0$:*

$$P\left\{\left|\frac{X_1 + ... + X_n}{n} - \mu\right| > \varepsilon\right\} \to 0, \quad when \;\; n \to \infty. \tag{2.10}$$

Its generalization is the following:

**Theorem 2.2.2** ***The strong law of large numbers.*** *Under the same conditions of the previous theorem, the following expression have probability equal to 1:*

$$\lim_{n\to\infty} \frac{X_1 + ... + X_n}{n} = \mu. \tag{2.11}$$

The Central Limit Theorem, is another very important probability to the Monte Carlo methods:

**Theorem 2.2.3** ***Central Limit Theorem*** *Let $X_1, X_2, ...$ be iid with mean $\mu$ and variance $\sigma^2$ it is true that:*

$$\lim_{n\to\infty} P\left\{\frac{X_1 + ... + X_n - n\mu}{\sigma\sqrt{n}} < x\right\} = \phi(x), \tag{2.12}$$

*where $\phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{\frac{-x^2}{2}} dx$ is a gaussian distribution for $-\infty < x < \infty$ .*

The integration with Monte Carlo method has its basis on above theorems, they guarantee the operation of Monte Carlo methods. To solve an intractable integral, a Monte Carlo method consists in generate iid samples under certain probability distribution $X_i \sim P(\cdot)$ considering the integral as the expected value of the function. The problem of Monte Carlo

[1] www.github.com/igomezv/IntroMCMC

methods is how to generate this independents samples, one solution to attack this issue is through the Markov Chains.

Figure 2.3 shows an example of Monte Carlo sampling that allows to calculate the integral of a function (blue line). Through random sampling and determining which sample is above or below the function it is possible to estimate the value of the integral.

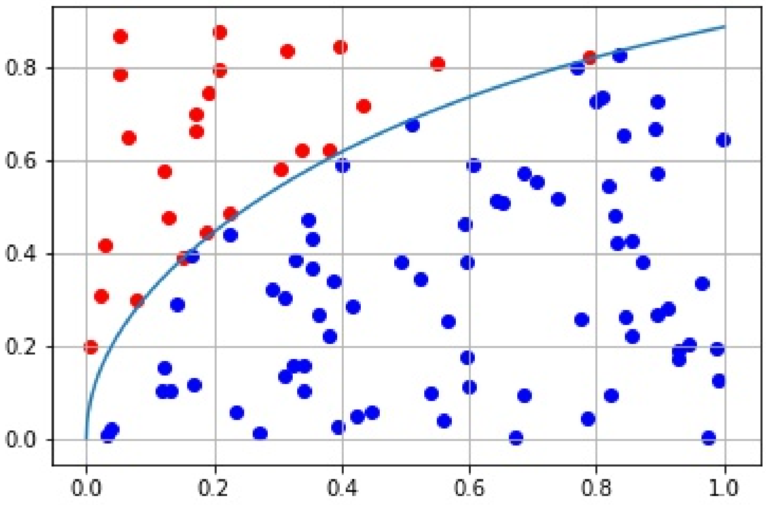


Figure 2.3: Monte Carlo sampling

## 2.2.4 Markov Chain Monte Carlo

A stochastic process is a collection of random variables $X_t : t \in T$, where $X_t$ takes values from a state space indexed by the set $T$, called time, which can be discrete or continuous. A stochastic process $X_n : n \in T$ is a Markov Chain if:

$$P(X_n = x|X_0, ..., X_{n-1}) = P(X_n = x|X_{n-1}) \tag{2.13}$$

therefore the probability of $X_n$ only depends of $X_{n-1}$. This condition is called Markov property. On the other hand, the transition probabilities are defined as:

$$p_{ij} \equiv P(X_{n+1} = j | X_n = i), \tag{2.14}$$

and the transition matrix $\mathbf{P}$ has entries $(i,j)$ as $p_{ij}$.

**Definition 2.2.1** *A Markov chain is irreducible if for each pair of states i and j there are a probability for the process to move from state i to state j.*

**Theorem 2.2.4** *An irreducible ergodic Markov chain has a unique stationary distribution $\pi$. The limiting distribution exists and is equal to $\pi$. If g is a bounded function, then with probability 1:*

$$lim_{N\to\infty}\frac{1}{N}\Sigma_{n=1}^{N}g(X_n) \to \mathbb{E}_{\pi}(g) \equiv \Sigma_j g(j)\pi_j \tag{2.15}$$

$\pi$ satisfies the *detailed balance* if:

$$\pi_i p_{ij} = p_{ji}\pi_j. \tag{2.16}$$

Therefore, we have the following:

**Theorem 2.2.5** *If $\pi$ satisfies detailed balance, then $\pi$ is a stationary distribution.*

Therefore, we need algorithms that generates a Monte Carlo sampling through Markov chains. The most popular algorithm of this type is Metropolis-Hastings:

The function $q(\cdot|X_t)$ is a distribution that can already be simulated (due to its symmetry, the normal distribution is usually chosen to facilitate the simulation process), $\pi(\cdot)$ is the objective function and $\alpha(X_t, Y)$ is defined as:

$$\alpha(X,Y) = min\left(1, \frac{\pi(Y)q(X|Y)}{\pi(X)q(Y|X)}\right). \tag{2.17}$$

In the original Metropolis-Hastings algorithm of 1953, it only uses symmetrical distributions where $q(Y|X) = q(X|Y)$, in this case the acceptance rate is:

$$\alpha(X,Y) = min\left(1, \frac{\pi(Y)}{\pi(X)}\right).$$

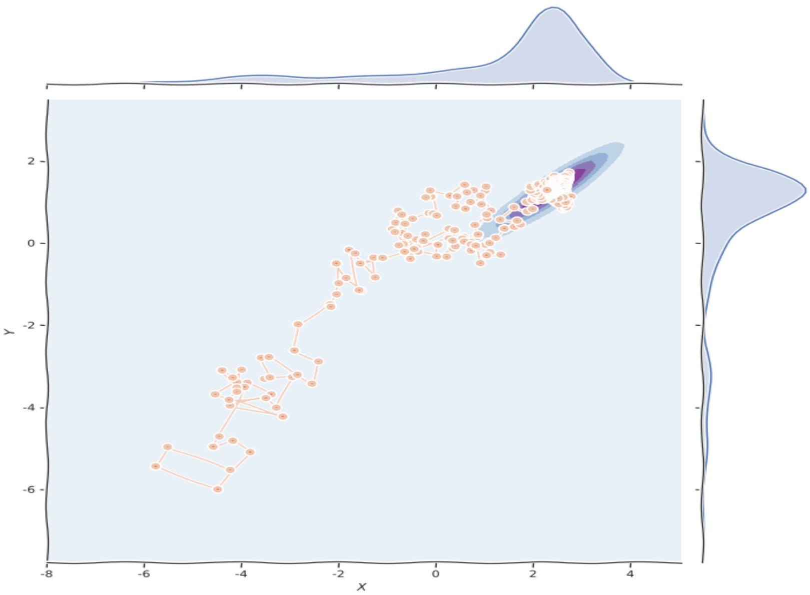


Figure 2.4: Random walk Metropolis-Hastings

**Algorithm 1:** Metropolis-Hastings

```
Initialise $X_0$, $nsamples$, $t = 0$;
while $t<nsamples$ do
    Generate a candidate $Y \sim q(\cdot|X_t)$;
    Generate $U \sim U(0,1)$;
    if $U \leq \alpha(X_t, Y)$ then
        $X_{t+1} = Y$
    else
        $X_{t+1} = X_t$
    end
    $t = t + 1$
end
```

On the other hand, the also popular random-walk Metropolis-Hastings algorithm uses $q(Y|X) = q(|X - Y|)$ as in the Figure 2.4. The correct choice of $\alpha(X,Y)$ allows that $\pi(\cdot)$ satisfies the deailed balance condition (Eq. 2.16 and that $\pi(\cdot)$ have the same stationary distribution of the Markov Chain.

In strictly words and as it can be seen in the above elements of the MCMC methods, there are not a convergence criterion. Therefore, there are two ways to stop the learning process, one is defining a limit number of samples generated. The second way is more formal and consists in verify if the Markov Chain have achieved a stationary state. One method to monitor this is the Gelman-Rubin test that consist in the following steps:

**Algorithm 2:** Gelman-Rubin diagnostic

```
Initialise N;
Initialise $M \geq 2$ number of Markov chains;
Generate $M$ Markov chains, each with $N$ steps;
R = 0.0;
while R<0.97 or R>1.03 do
    Generate $N$ new samples for each Markov chain;
    Burn the the first $N$ iterations in each chain ;
    # $s^2$ variance of each chain:;
    $W = \frac{1}{M}\Sigma_{j=1}^{M} s_j^2$ ;
    # Calculate the variance between chains;
    $B = \frac{N}{M-1}\Sigma_{j=1}^{M}(\bar{\theta}_j - \bar{\bar{\theta}})^2$;
    # $\bar{\bar{\theta}}$ is the mean of the $M$ chains;
    $var(\theta) = (1 - \frac{1}{N})W + \frac{1}{N}B$ ;
    $R = \sqrt{\frac{var(\theta)}{W}}$;
```

### 2.2.5 Nested Sampling

Nested sampling is a category of Bayesian inference algorithms to estimate the Bayesian evidence, with its uncertainty, and as by-product sampling the posterior probability density function.

The basic idea of nested sampling is to simplify the integral of Bayesian evidence by mapping the parameter space in an unit hypercube. The fraction of the prior contained within an iso-likelihood contour $\mathscr{L}_c$ in the unit hypercube is called called prior volume:

$$X(\mathscr{L}) = \int_{\mathscr{L}(\theta)>\mathscr{L}_c} \pi(\theta)d\theta. \tag{2.18}$$

The Bayesian evidence can be reduced as an one-dimensional integral of the Likelihood as function of the prior volume $X$:

$$Z = \int_0^1 \mathscr{L}(X)dX. \tag{2.19}$$

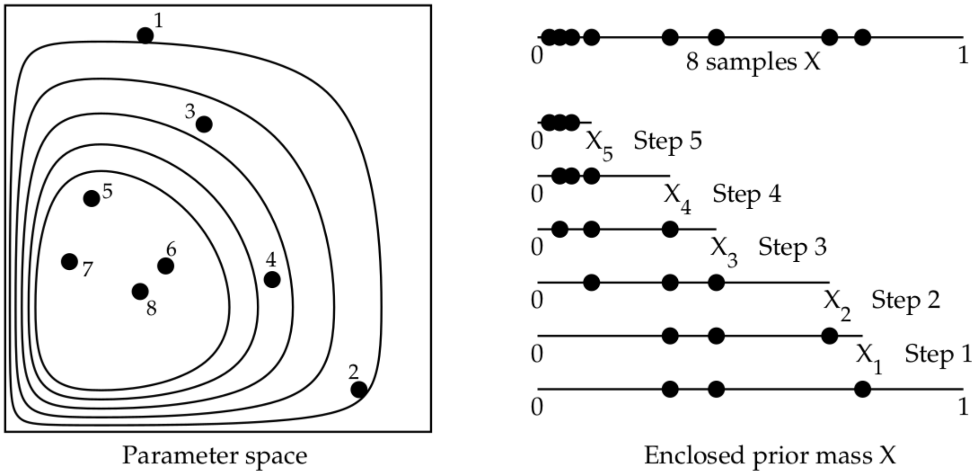


Figure 2.5: Five steps of nested sampling with three live points. Source image: [1]

Nested sampling starts with a specific number $n_{live}$ of random points, called live points, within the prior volume given by the constrained prior. This samples are ordered according their likelihoods values. In each new iteration, the worst point (with the lowest likelihood $\mathscr{L}_{worst}$) is removed (see Figure 2.5). A new sample is generated within a contour delimited by $\mathscr{L}_{worst}$ and with a likelihood $\mathscr{L}(\theta) > \mathscr{L}_{worst}$. It is expected that the prior volume at each iteration be compressed by a factor $t$, that in the *crude* implementation is:

$$t = e^{-1/n_{live}}, \tag{2.20}$$

and the Eq. (2.19) can be simplified as a Riemann sum:

$$Z \approx \sum_{i=1}^{N} L_i \omega_i, \tag{2.21}$$

where $\omega_i$ is the difference between the prior volume of two consecutive points: $\omega_i = X_{i-1} - X_i$.

At every moment, nested sampling maintains the population of $n_{live}$ live points and the final set of live points are agglomerated in a zone of high probability.

According the way that the sampling from the constrained prior is performed, there are different nested sampling algorithms. For example, `MultiNest` [16] uses rejection sampling within ellipsoids, while in `Polychord` [17] the points are generated with slice sampling.

**Algorithm 3:** Nested Sampling (crude implementation) from [18]

```
Generate $N$ from samples prior PDF;
Initialise $Z = 0, X_0 = 1, it = 0, maxit$;
while $it < maxit$ do
    Record the lowest of the current likelihood values as $L_i$;
    $X_i = exp(-1/N)$;
    $w_i = X_{i-1} - X_i$;
    $Z = Z + L_i * w_i$;
    replace point of lowest likelihood by new one sample from within $L(\theta) > L_i$;
    it += 1;
$Z = Z + N^{-1}(L(\theta_1) + ... + L(\theta_N))X_{maxit}$;
```

The Figure 2.6 shows a nested sampling in different stages. Unlike the random walk of Metropolis Hastings, each new sample is within the previous worst likelihood contour.

Figure 2.7 shows the main difference between a MCMC algorithm like Metropolis-Hastings and a nested sampling algorithm.

For more details about nested sampling we recommend the references [1, 17, 19].

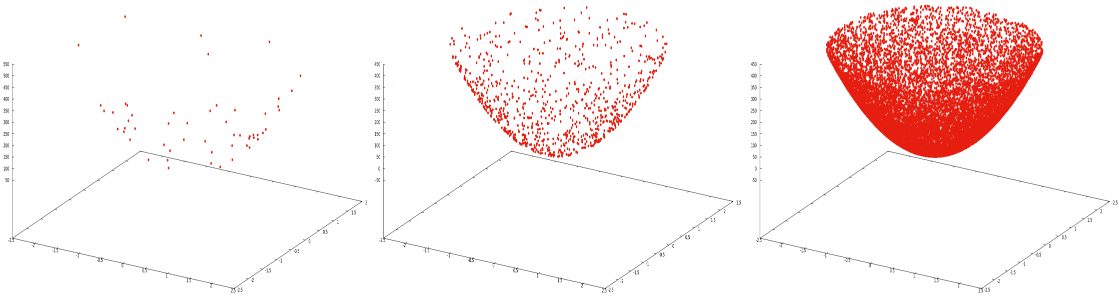


Figure 2.6: Three different stages of a nested sampling

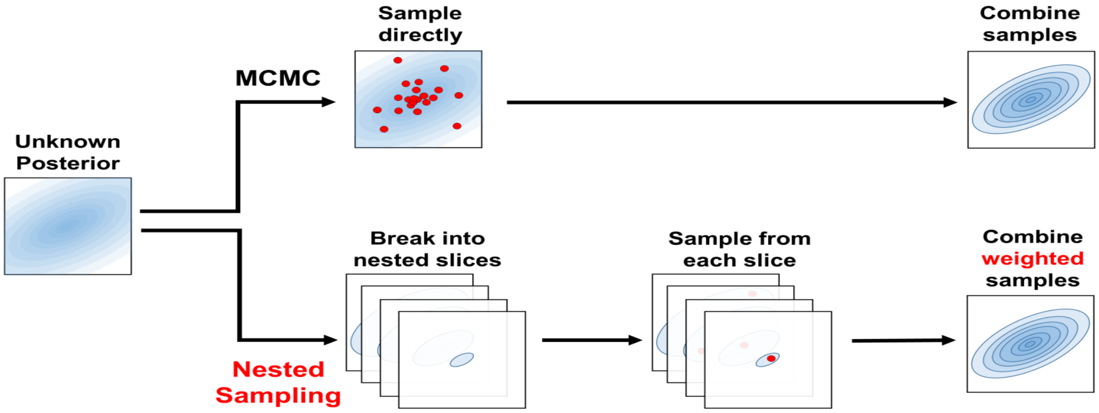


Figure 2.7: MCMC sampling vs nested sampling

## 2.3 Non-parametric inference

There are situations in which there are not a functional way to describe the data or the information about the data is very poor. In these cases, non-parametric inference to analyze the data with the fewest possible assumptions. In this work we do not use these methods, rather we use Artificial Neural Networks to make non-parametric inferences, however, we believe that a brief description about it could be useful. For more details on these types of methods, see Reference [7, 20].

A non-parametric model is a set of statistical models that cannot be parametrized by a finite number of parameters and allows "free form" solutions. To estimate in a non-parametric way a probability density function it is necessary to sort some smoothness assumptions on the data. A popular method is the histogram as a density estimator, which divides the real line in small parts called bins; this type of estimator is a piecewise constant function and the number of observation in each bin is proportional to the height of the this function.

Other nonparametric methods are known as kernel density estimators. In these cases, a kernel function is centered on each data point and if this function is smooth, then the result will be a smooth density estimate, unlike the histogram method where there is a great dependence on the number of bins and their width. Under these type of methods are Gaussian processes and predictive models.

To test a nonparametric model, it is necessary to measure how far its predictions are from the expected values. Let $X_1, ...., X_n$ be $n$ iid data points from some distribution. We define a point estimator $\hat{\theta}_n$ of a parameter $\theta$ as a function $\hat{\theta}_n = g(X_1, ..., X_n)$. A common metric is Mean Squared Error (MSE) defined as follows:

$$MSE = \frac{1}{n}\sum_{i}^{n}(\theta_i - \hat{\theta}_i)^2, \tag{2.22}$$

where $\theta_i$ is a vector with predictions, $\hat{\theta}_i$ is a vector with the expected values and $n$ is the number of predictions (or the length of $\theta_i$ and $\hat{\theta}_i$).

**Theorem 2.3.1** *The MSE can be written as follows:*

$$MSE = bias^2(\hat{\theta}) + variance(\hat{\theta}), \tag{2.23}$$

*where* $bias = \mathbb{E}(\hat{\theta}) - \theta$.

The bias measures how far the neural network predictions are from the actual value, while variance refers to how much the prediction varies at nearby points. As the complexity of the model increases, the bias can decrease and the variance can increase, this is called the bias-variance dilemma [21]. A model with high variance will be overfitted, while a model with high bias will be insufficient to learn the complexity of the data (underfitting). In both cases, the model generated by the non-parametric have inaccurate predictions.

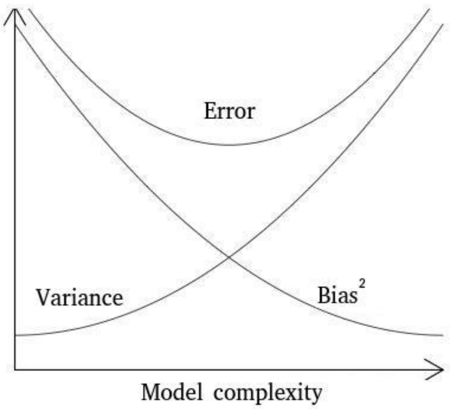


Figure 2.8: Bias and variance related with model complexity

## 2.4 Artificial neural networks

Artificial neural networks are computational models that was inspired on the biological neurons (see Fig. 2.9). They had been idealized in the 1940s, but due to the limitations of computing power, their development was successful until 1980s. In recent years, with parallel computing and other advances in Computer Science, the ANNs have had a new resurgence. They are part of the tools used in machine learning. In fact, the perceptron (the simplest type of ANN) was the first learning machine.

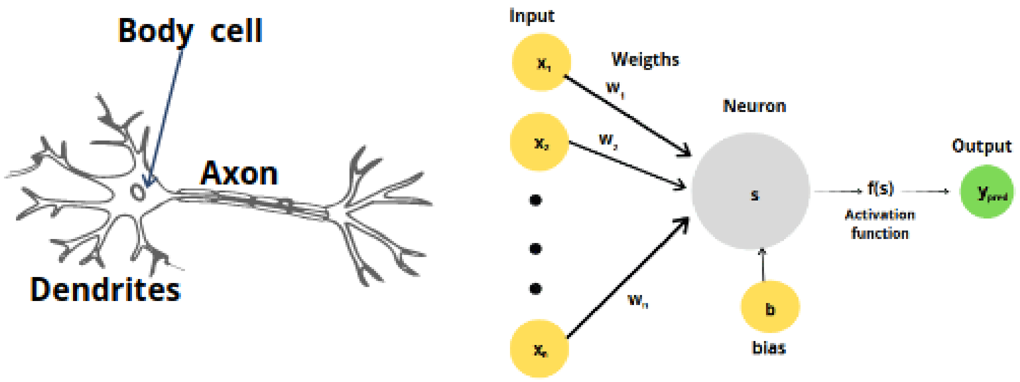


Figure 2.9: Analogy between a biological neural networks and an ANN (perceptron)

As is common in supervised machine learning, at the beginning of the ANN training, the original dataset is separated in two parts: training and validation sets. An usual choice is 80% and 20% respectively. The first set is used to train the ANN, while the validation set contains unseen values, therefore it is useful for testing the performance of the ANN and evaluating its ability to produce a good model to the input dataset. Figure 2.10 shows a diagram of this process.

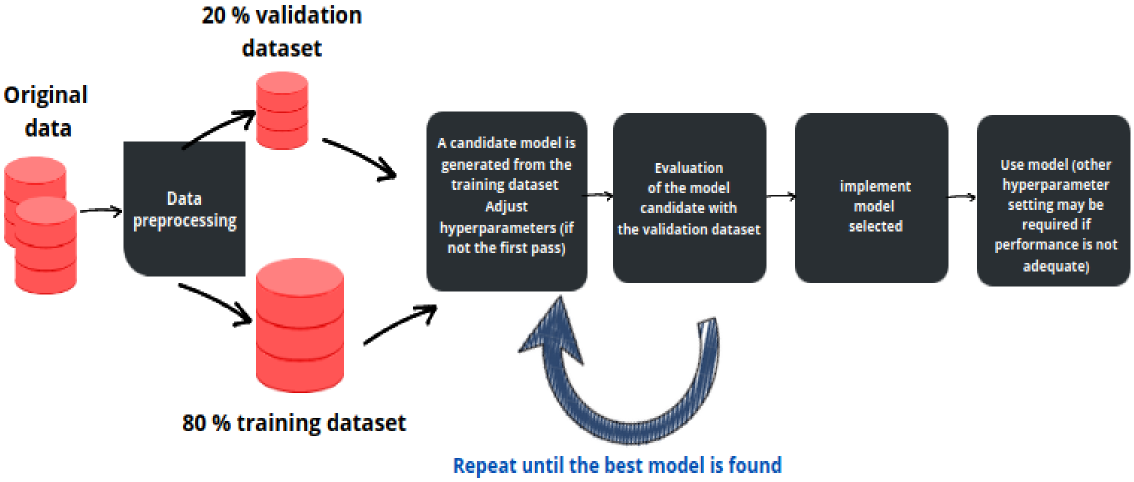

20 % validation dataset
Original data
Data preprocessing
A candidate model is generated from the training dataset Adjust hyperparameters (if not the first pass)
Evaluation of the model candidate with the validation dataset
implement model selected
Use model (other hyperparameter setting may be required if performance is not adequate)
80 % training dataset
Repeat until the best model is found

Figure 2.10: Flow of supervised machine learning

On the other hand, the *Universal Approximation Theorem* [22] states that an ANN with at least one hidden layer with a finite number of neurons can approach any continuous function if the activation function is continuous and nonlinear. Figure 2.11 shows some examples of these types of activation functions.

In general, an ANN is a directed graph with the following properties:

- Every node $i$ has a stage variable $x_i$
- Every connection $(i, j)$ of $i, j$ nodes has an weight $w_{ij} \in \mathbb{R}$.
- Each node is associated with an threshold $\theta_i$.
- For each node $i$ a function $f_i(x_j, w_{ij}, \theta_i)$ is defined that depends on the weights of its connections, the threshold and the states of the nodes j to it connected. This function

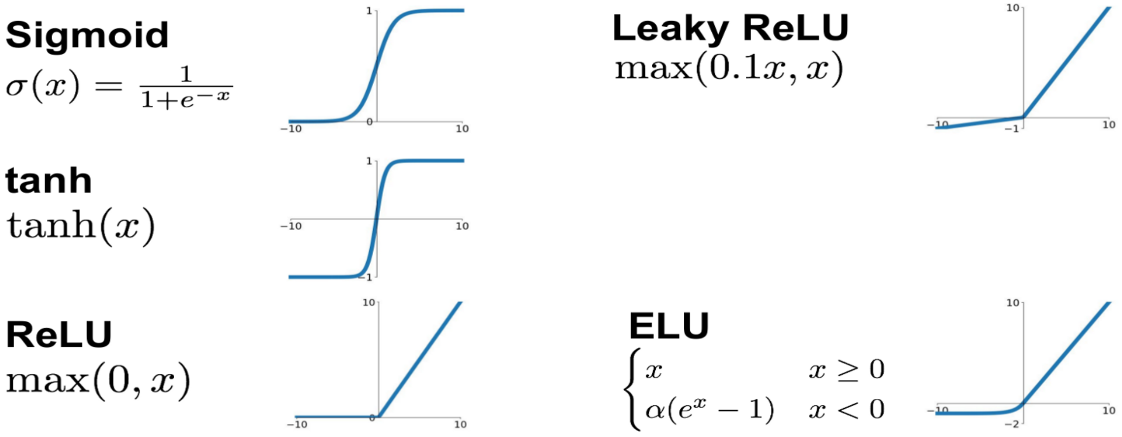


Figure 2.11: Examples of activation functions

provides the new state of the node.

According to the above, the nodes are the neurons and the weights the synapses. Therefore, a neuron without incoming synapses is an input neuron and a neuron without outgoing synapses is an output neuron. Intermediate neurons are called hidden neurons and form hidden layers.

If the network does not present closed loops of connections, it is unidirectional or feedforward. In counterpart, if there exists feedback, then it is a recurrent network.

In general, the learning mechanism of an ANN is as follows:

- The first layer of neurons reads the features of the dataset. In each connection between a neuron and another is assigned a random number (we use random numbers with a normal distribution centred on 0 with a standard deviation of 0.01). The input data make up a matrix $X_1$ and provides the values for the first layer of nodes. $X_i$ refers to the values of the nodes in the *i-th* layer. The weights make up another matrix $W_i$ and they are the values for the connections between the *i-th* and the *(i+1)-th* layers. The product $Z$ of these two matrices is the following:

$$Z_{i+1} = W_i^T X_i, \tag{2.24}$$

where $W_i \in \mathbb{R}^{m \times n}$, with $m, n$ as the number of nodes in the *i-th* and *(i+1)-th* layers respectively. $X_i$ corresponds to the *i-th* layer, therefore has $m$ dimensions. It is worth to apply the transpose of $W_i$ in order to allow the matrix product.

- A nonlinear activation (or transfer) function $\phi$ modulates $Z_i$ and assigns values to the next layer of neurons. This process, known as forward propagation, is repeated until the last layer is reached. The values of neurons in subsequent layers are given by:

$$X_{i+1} = \phi(Z_{i+1}). \tag{2.25}$$

- The value of the neurons in the last layer must be evaluated by an error function (or loss function) which measures the difference between the value given by the ANN and the expected one. The loss function is minimised by an optimisation algorithm, typically such as *gradient descent* combined with the *backpropagation* algorithm [23, 24] to calculate gradients. In this paper we use the Mean Squared Error (MSE) as a loss function because it is the usual selection in regression problems. The goal of the training in an ANN is to minimise the loss function.

- During backward propagation the weights are updated, then forward propagation is performed again. This is repeated until the loss function reaches the desired precision and then the neural network is trained and ready to make predictions.

### 2.4.1 Hyperparameters

The internal parameters of a neural network model are known as hyperparameters. Some of them, such as the number of nodes and layers, have already been defined above. Also considered hyperparameters are the algorithm used to minimize the loss function, as well as the loss function, since there is a considerable range of options for each case. In the following list, we will describe others that we used in this work.

- The number of samples propagated through the network before updating the weights is known as batch size.

- Each iteration of the entire data set constitutes an epoch.

- The dropout (DO), is a regularisation technique [25] that allows smaller values to be achieved in the loss function and prevents overfitting. It consists of randomly turning off neurons during training, so the neurons that operate at each epoch are different. The associated hyperparameter is a scalar value that indicates the probability of turning off a neuron in each epoch.

As previously mentioned, ANNs have the ability to approximate any function, in other words, they can generate computational models that generalise the input data. It is worth to select carefully a good combination of them to guarantee that the ANN model has the capability of generalisation, an incorrect choice of them can produce undesirable models, either underfitted or overfitted with respect to the data. A neural network well trained should satisfy the bias-variance dilemma of the Equation 2.23.

There are several approaches to tune the hyperparameters [26–29]. In this work, for simplicity, we use a common empirical strategy based on a grid of hyperparameters [29]. Figure 2.12 shows an example grid in which four hyperparameters are varied to find the best combination of them (those that minimize the loss function).

The hyperparameter grid consists of a selection of possible values for all the ANN parameters to be adjusted. The ANN is trained for all combinations of them, included on the grid, to find the one that obtains the lowest value for the loss function in the validation set. In addition, it is necessary to verify that the loss function of both the validation set and the training set have convergent behaviour to ensure that the ANN model is well trained (neither overfitting nor underfitting).

Whenever overfitting occurs it is mainly because the behaviour of the loss function in the test and training sets shows that the model has a high error in its predictions and, therefore, also has a high bias. In the other case, overfitting occurs when the loss function evaluated at the validation shows an increasing trend, or because there is a considerable gap with the loss function of the training set. We use the difference between the predictions of the last two epochs ($\Delta MSE_{val}$) to get an idea about the variance of the ANN model, the smaller this error, the smaller the variance.

### 2.4.2 Feedforward neural networks

The Feed-forward Neural Network (FFNN)s, also called multilayer perceptrons or deep feedforward networks, are the quintessential deep learning models [30]. In this type of ANN the connections between layers and the information flow are straight forward. They are composed of one input layer, at least one hidden layer and an output layer. The input is conformed by the independent variables (or features) of the dataset, while the output is the dependent variables (or labels).

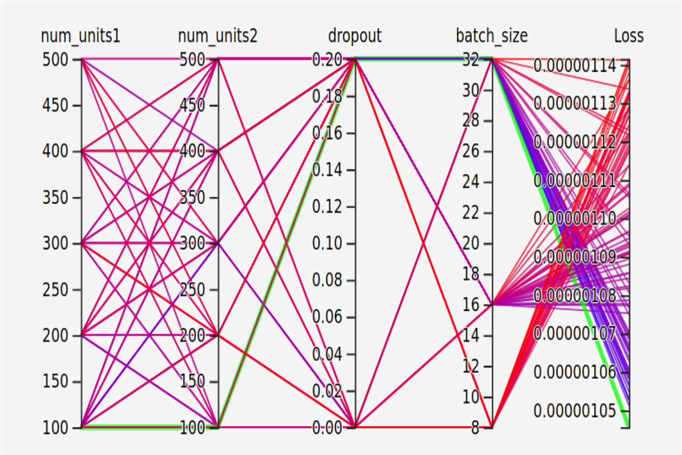


Figure 2.12: Hyperparameter grid

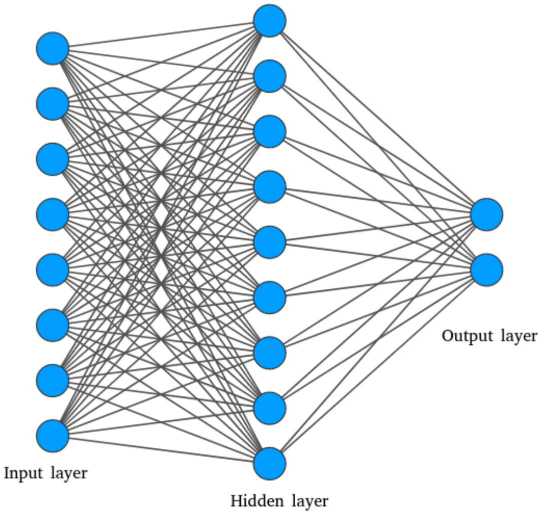


Figure 2.13: Feedforward Neural Network (FFNN)

### 2.4.3 Autoencoder

The other type of ANN used in this work is the Autoencoder (AE) [31], that is trained to generate a copy of its input on its output. These type of neural networks can be thought as two symmetrical coupled ANNs, where the first (encoder) makes a dimensional reduction for the input and obtains a coded representation (vector embedding) of the original data. The second part (decoder) takes the coded representation of the data and recovers an instance with the same dimension of the original input. The encoder is a function $f$ that maps the input $x$ with dimension $l$ to an encoded vector $h$ with dimension $m$, with $m < l$:

$$f : x \in \mathbb{R}^l \to h \in \mathbb{R}^m, \tag{2.26}$$

where $h_i := f_i(x) = \phi(W_i^T X_i),\ i = 1, 2, ..., m$ with $\phi$ as activation function. The decoder is the following $g$ function, that maps the encoded representation with dimension $m$ into an output $\hat{x}$ with the same dimension $l$ as the original input $x$:

$$g : h \in \mathbb{R}^m \to \hat{x} \in \mathbb{R}^l. \tag{2.27}$$

If the activation function, used in the autoencoder, is the identity function, i.e. $\phi(x) = x$, therefore this type of neural network is analogous to the Principal Component Analysis (PCA) technique. The figure 2.14 is an example of an autoencoder.

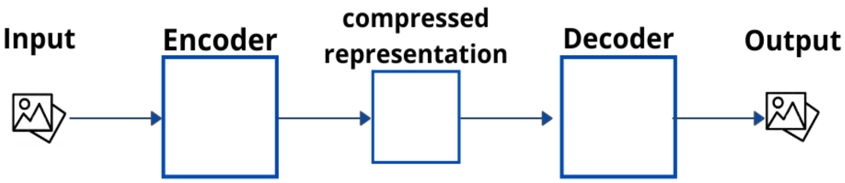


Figure 2.14: Classical autoencoder has an encoder, a decoder and a discrete compressed representation.

### 2.4.4 Variational Autoencoder

Here we briefly describe the idea of a variational autoencoder (VAE) [32, 33], however for an extended review see [34, 35]. In addition, we explain a first approach method used to generate synthetic covariance matrices from the original covariance matrix of the JLA SNeIa binned version.

Variational autoencoders use variational inference to sample the compressed representation (or latent space) and, therefore, allow to know the probability density function associated, precisely, to the compressed representation. Unlike classical autoencoders, such as those described earlier in this work, two layers of the same dimension as the latent space are designed before the compressed representation, whose function is to generate values to sample the mean $\mu$ and variance $\sigma$ which are the parameters of the statistical distribution that produces an input data (matrix or image) of the VAE to generate a point $z$ of the latent space.

As a way to construct a latent space distribution similar to the proposed Gaussian distribution, the Kullback-Leiber divergence (KL) is used. Thus, the selection of the relevant loss function to train the VAE is as follows:

$$loss_{\mathrm{VAE}} = \mathrm{MSE} + \mathbf{KL}(q(z|x)||p(z)) \tag{2.28}$$

where $q(z|x)$ is the probability density function to generate a $z$ point of the latent space given an input $x$. On the other hand, we can assume that $p(z) = N(0, I)$ with $p$ a probability density function of the $z$ points in latent space and $N$ a normal distribution centered at 0 with covariance matrix equal to the identity matrix. Because VAEs are widely used in image processing, it is more common to choose *binary cross entropy* [36] instead of MSE, however our interest is in the numerical information and not in a classification problem that takes place in image generation.

A diagram of a variational autoencoder is shown in the Figure 2.15, where the VAE has a continuous latent space that is sampled with the mean and variance layers; the notation is the same as in Equation 2.28 with $z$ the latent space variable, $x$ is the input variable and the encoder and decoder are associated with conditional probability density functions.

### 2.4.5 Monte Carlo Dropout

Due to its random nature, the dropout (schematized in Figure 2.16) can be used as a Monte Carlo simulation. When an ANN is trained, the dropout can be implemented in such a way that each prediction is different because the active neurons are different at each epoch. Therefore, it is possible to make several predictions, and thus obtain the average and standard deviations. Using this formalism, dubbed Monte Carlo dropout (MC-DO) [37] we can obtain a statistical uncertainty of a trained ANN model. We apply the dropout method to the FFNNs implemented in this work and compare the results with those solely with

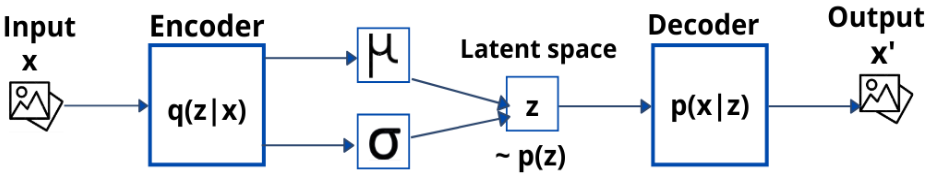


Figure 2.15: Variational Autoencoder

FFNNs.

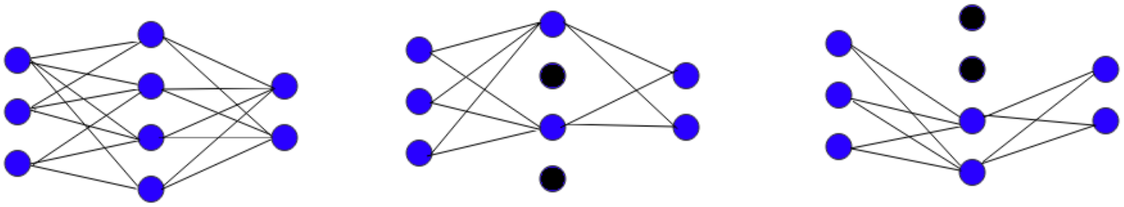


Figure 2.16: The dropout regularization technique turns off random neurons at each epoch.

# CHAPTER 3

# COSMOLOGICAL FRAMEWORK

## 3.1 Introduction

Cosmology is the study of the Universe on very large scales as a whole. At this scale a complex galaxy is a simple dot, and it is valid the Cosmological Principle that is the notion that the spatial distribution of matter, at large scales, in the Universe is homogeneous and isotropic. Homogeneity refers to the fact that all the places in the universe are equivalent, that is, that any observer will measure the same physical properties. On the other hand, the isotropy makes reference to that, in addition, each observer will measure the same in all the directions that observe. Along this manuscript the cosmological models, functions and datasets use the units $\hbar = c = 8\pi G = 1$.

### 3.1.1 General Relativity

The General Theory of Relativity (GTR) is the gravitational theory that governs the evolution of the Universe and a fundamental element of this theory is the metric tensor $g_{\mu\nu}$. This tensor has symmetry:

$$g^{\mu\nu}g_{\nu\sigma} = g_{\lambda\sigma}g^{\lambda\mu} = \delta^{\mu}_{\sigma}, \tag{3.1}$$

where $\delta^{\mu}_{\sigma}$ is the delta of Kronecker and the symmetry of the metric tensor implies that its inverse $g^{\mu\nu}$ is also symmetric.

The metric tensor $g_{\mu\nu}(x^{\alpha})$ describes the spacetime geometry and determines the invariant proper distance $ds$ between two events. We can note:

$$ds^2 = g_{\mu\nu}(x)dx^\mu dx^\nu \tag{3.2}$$

$$\tag{3.3}$$

$$\begin{aligned} g_{\mu\nu}(x)dx^\mu dx^\nu &= g'_{\alpha\beta}(x')dx'^\alpha dx'^\beta \\ &= g'_{\alpha\beta}(x')(\frac{\partial x'^\alpha}{\partial x^\mu}dx^\mu)(\frac{\partial x'^\beta}{\partial x^\nu}dx^\nu) \\ &= \frac{\partial x'^\alpha}{\alpha x^\mu}\frac{\partial x'^\beta}{\partial x^\nu}g'_{\alpha\beta}(x')dx^\mu dx^\nu, \end{aligned} \tag{3.4}$$

where we have a general coordinate transformation $x^\alpha \to x'^\alpha$.

The spacetime is a 4-dimensional pseudo-Riemannian manifold $(M, g_{\mu\nu})$ with a differentiable manifold $M$ and a metric $g_{\mu\nu}$. In a curved manifold, the trajectory of a particle is given by geodesics:

$$\frac{d^2x^\mu}{ds^2} + \Gamma^\mu{}_{\nu\rho}\frac{dx^\nu}{ds}\frac{dx^\rho}{ds}, \tag{3.5}$$

where $\Gamma^\alpha{}_{\beta\gamma}$ are the Christoffel symbols defined as following:

$$\Gamma^\rho{}_{\mu\nu} = \frac{g^{\rho\sigma}}{2}(\frac{\partial g_{\nu\sigma}}{\partial x^\mu} + \frac{\partial g_{\sigma\mu}}{\partial x^\nu} - \frac{\partial g_{\mu\nu}}{\partial x^\sigma}), \tag{3.6}$$

where $x^\mu$ are local coordinates. Geodesics followed by massive particles are assumed to be time-like, whereas massless particles (e. g. photons) move along null-like geodesics. A particle that moves along space-like geodesics does not have physical sense because thy propagate at superluminal speeds.

The Riemann tensor have information about the curvature of the spacetime, given by:

$$R^\rho{}_{\sigma\mu\nu} = \partial_\mu\Gamma^\rho{}_{\nu\sigma} - \partial_\nu\Gamma^\rho{}_{\mu\sigma} + \Gamma^\rho{}_{\mu\lambda}\Gamma^\lambda{}_{\nu\sigma} - \Gamma^\rho{}_{\nu\lambda}\Gamma^\lambda{}_{\mu\sigma}. \tag{3.7}$$

We can contract the first and third indices of the Riemann tensor and we can find the Ricci tensor: $R_{\sigma\nu} = g^{\rho\mu}R_{\rho\sigma\mu\nu}$. Then, contracting again the indices, we obtain the Ricci scalar: $R = g^{\mu\nu}R_{\mu\nu}$.

Einstein's field equation (the equation of GTR) indicates how the metric responds to the matter in it through its energy and its momentum, hence it contains the dynamics of the gravitational field:

$$R_{\mu\nu} - \frac{1}{2}Rg_{\mu\nu} = 8\pi G T_{\mu\nu} \tag{3.8}$$

where $T_{\mu\nu}$ is the energy-momentum tensor of the matter fields. We are assuming units such that the speed of light is $c = 1$. The dynamical evolution of the metric $g_{\mu\nu}$ according to the dynamical changes of the matter fields represented by $T_{\mu\nu}$ is described by (3.8).

The Eq. (3.8) can be seen as a relationship between the geometry of the spacetime (left side of equation) and its energy or physics (right side). This equation tells us how the curvature of spacetime reacts to the presence of energy-momentum.

One can use the variational approach of the field equations, Eq. (3.8). Then, using the Einstein-Hilbert action:

$$S = \int d^4x\sqrt{-g}\Big[\frac{1}{16\pi G}R + S_m\Big]. \tag{3.9}$$

It is the most general action containing up to two derivatives of the metric, guaranteeing that the field equation contains up to second orders of the metric. If we vary the action (3.9) with respect to the metric, we get (3.8).

### 3.1.2 Cosmology

Mathematically, to guaranty the Cosmological Principle, we need a geometry that allow it, and an isotropic and homogeneous manifold is given by the Friedman-Lemaître-Robertson-Walker (FLRW) metric:

$$ds^2 = -dt^2 + a^2(t)\Big(\frac{dr^2}{1-kr^2} + r^2(d\theta^2 + sin^2\theta d\phi^2)\Big) \tag{3.10}$$

where $a(t)$ is the scale factor, $k$ is the curvature parameter ($k = 1$ for closed universes, $k = 0$ for flat and $k = -1$ for open universes). For the following equations, we adopt units such that $8\pi G = 1$. It is worth mentioning that the scale factor $a(t)$ indicates the growth of the relative distance between the comoving points in the spacetime. Hence, the dynamical evolution of the Universe as a whole is dictated by $a(t)$. The functional form of $a(t)$ can be found by solving the Eqs. (3.8) with the input (3.10).

To completely determine how the Universe evolves we must choose a metric, in this case the metric of the Eq. 3.10 and solve Eq. 3.8. To do that let us assume that the Universe is dominated by a perfect fluid in its own rest-frame with a 4-velocity given by:

$$u^\mu = (1,0,0,0), \tag{3.11}$$

and the following energy-momentum tensor:

$$T_{\mu\nu} = (p+\rho)u_\mu u_\nu + pg_{\mu\nu} = \begin{bmatrix} \rho & 0 & 0 & 0 \\ 0 & & & \\ 0 & & g_{ij}p & \\ 0 & & & \end{bmatrix}, \tag{3.12}$$

where $p$ is the pressure and $\rho$ the energy density of the fluid.

The Equation of State (EoS) of perfect fluids is given by the ratio of their pressure and energy density:

$$\omega = \frac{p}{\rho} \tag{3.13}$$

and the conservation of energy equation becomes:

$$\frac{\dot{\rho}}{\rho} = -3(1+\omega)\frac{\dot{a}}{a} \tag{3.14}$$

Different kind of fluids or matter species (see Section 3.1.3) are present in the Universe, each of them with its own EoS. The energy-momentum tensor incorporates all of the components in the Universe so it simply amounts to the sum of multiple terms like (3.12), each with its specific value of $p$ and $\rho$. For future purposes, it is worth recalling that ordinary perfect fluids fulfill the Strong Energy Condition:

$$\rho + 3p \geq 0. \tag{3.15}$$

Employing the FLRW metric (3.10), Christoffel symbols, Ricci Tensor and Ricci Scalar can be computed and inserted in the Field Equation, (3.8). By solving the time-time component $G_{00}$ and the space-space components $G_{ij}$ we obtain the Friedmann Equations (FE), that, following the evolution of a(t), describe the dynamic of the Universe as whole. These equations read:

$$H^2 \equiv (\frac{\dot{a}}{a}) = \frac{1}{3}\rho - \frac{\kappa}{a^2}, \tag{3.16}$$

$$\dot{H} + H^2 = \frac{\ddot{a}}{a} = -\frac{1}{6}(\rho + 3p), \tag{3.17}$$

where $\dot{x}, \ddot{x}$ represent derivative and double derivative respect to the time $t$ and $H$ is the Hubble parameter. These equations are known as Friedmann equations and they can be combined into the continuity equation as following:

$$\begin{aligned}\frac{d\rho}{dt} &= -3H(\rho+p) \qquad (3.18)\\ &\text{or}\\ \frac{d(ln\rho)}{d(lna)} &= -3H(1+\omega). \qquad (3.19)\end{aligned}$$

Integrating Eq. (3.19) and using Eq. (3.16) leads to the solution for the scale factor:

$$a(t) \propto \begin{cases} t^{\frac{2}{3(1+\omega)}} & if \quad \omega \neq -1 \\ \\ e^{Ht} & if \quad \omega = -1 \end{cases} \qquad (3.20)$$

This shows that the qualitative behavior of the cosmological evolution depends crucially on the equation of state $\omega$.

To complement the basic terminology of cosmological parameters, it is worth mentioning the rate of expansion of the Universe characterized by the Hubble parameter:

$$H = \frac{\dot{a}}{a}. \qquad (3.21)$$

It is also important to define the deceleration parameter:

$$q = -\frac{a\ddot{a}}{\dot{a}^2}, \qquad (3.22)$$

which measures the rate of change of the rate of expansion.

Another useful quantity, and very important in the test of cosmological models, is the density parameter:

$$\Omega = \frac{8\pi G}{3H^2}\rho = \frac{\rho}{\rho_c}, \qquad (3.23)$$

where $\rho_c$ is the critical density defined by:

$$\rho_c = \frac{3H^2}{8\pi G}. \qquad (3.24)$$

The critical density changes with time and is called *critical* because the Friedman equation (Eq. 3.16) can be written:

$$\Omega - 1 = \frac{\kappa}{H^2a^2}, \qquad (3.25)$$

and we have the following:

$$\text{if}\begin{cases}\rho<\rho_c & \Longrightarrow \Omega<1 \Longrightarrow \kappa<0 \quad \Longrightarrow \text{ open universe}\\ \\ \rho=\rho_c & \Longrightarrow \Omega=1 \Longrightarrow \kappa=0 \quad \Longrightarrow \text{ flat universe}\\ \rho>\rho_c & \Longrightarrow \Omega>1 \Longrightarrow \kappa>0 \quad \Longrightarrow \text{ closed universe}\end{cases}$$

Therefore, the density parameter tells us which of the three FLRW geometries describes our Universe. Observational data are of crucial importance to know its value; currently, the CMB data allows us to believe that $\Omega \simeq 1$, so a flat universe is the most feasible option.

For a homogeneous universe filled with matter or radiation, GRT predicts that the cosmic expansion will slow down over time [38]. In the 1990s, however, two independent studies of supernovae shown that the expansion of the universe has accelerated over the last billions of years [39, 40].

One of the major challenges of the cosmology is, precisely, explain the acceleration of the Universe and the best candidate is a mysterious component called Dark Energy (DE).

### 3.1.3 The components of the Universe

The matter species of the Universe are broadly classified into relativistic particles, non-relativistic matter and dark energy. The relatvistic particles correspond to photons (bossons) and some neutrinos (fermions) and their EoS without degeneracies is given by $\omega = \frac{1}{3}$. Particularly, the photons have an energy density and density parameter are the following (based on the CMB results):

$$\rho_{\gamma 0} = 4.641\times 10^{-34} gcm^{-3}, \tag{3.26}$$

$$\Omega_{\gamma 0} = \frac{8\pi G\rho_{\gamma 0}}{3H_0^2} = 2.469\times 10^{-5} h^{-2}. \tag{3.27}$$

On the other hand, the energy density of the neutrinos and anti-neutrinos (together) is given by:

$$\rho_\nu = N_{eff} = \frac{7\pi^2}{120} T_\nu^4, \tag{3.28}$$

where $N_{eff}$ is the effective number of neutrino species and $T_\mu$ is its temperature. Then, the density parameter of radiation, that considers the sum of photons and relativistic neutrinos, is:

$$\Omega_{r0} = \frac{\rho_{\gamma 0} + \rho_{\nu 0}}{\rho_{c0}} = \Omega_{\gamma 0}(1 + 0.2271 N_{eff}). \tag{3.29}$$

Using the accepted values $h = 0.72$ and $N_{eff} = 3.04$, we obtain $\Omega_{r0} = 8.051 \times 10^{-5}$.

In the case for non-relativistic particles, there are baryons that their density parameter has been estimated from observations of BAO, WMAP and SNe-Ia:

$$\Omega_{b0} h^2 = 0.02267^{+0.00058}_{-0.00059}. \tag{3.30}$$

The baryonic matter alone, however, is not sufficient to allow a consistent structure formation with the observations of large scale structure, hence is necessary to considerate the existence of dark matter as another non-relativistic component in the Universe. In fact, the CMB anisotropy data show that the present abundance of dark matter is about 5 times larger than the baryons. The data of WMAP [41] constrain the density parameter of CDM (Cold Dark Matter) as following:

$$\Omega_{c0} h^2 = 0.1131^{+}_{-}0.00034. \tag{3.31}$$

Despite observations confirm the existence of dark matter, its origin has not been identified yet.

If we sum the density parameters of radiation, baryons and dark matter, we can note that the result does not exceed 0.3 and, precisely, the Dark Energy is the unknown component that corresponds to the remaining 70% of the cosmic matter. The observational data from WMAP, SNe-Ia and BAO constrain the present density parameter of DE:

$$\Omega_{DE0} = 0.726 \pm 0.015. \tag{3.32}$$

The standard cosmological model explains the Dark Energy by the cosmological constant and the Einstein equation becomes:

$$R_{\mu\nu} - \frac{1}{2} R g_{\mu\nu} + \Lambda g_{\mu\nu} = 8\pi G T_{\mu\nu} \tag{3.33}$$

which can be derived from the action:

$$S = \frac{1}{16\pi G} \int d^4x \sqrt{-g}(R - 2\Lambda) + S_m \tag{3.34}$$

The Friedmann equation describing the late-time dynamical evolution for a flat-ΛCDM model can be written as

$$H(z)^2 = H_0^2\left[\Omega_{m,0}(1+z)^3 + (1-\Omega_{m,0})\right], \tag{3.35}$$

where $H$ is the Hubble parameter and $\Omega_m$ is the matter density parameter; subscript 0 attached to any quantity denotes its present-day $(z=0)$ value. In this case, the EoS is $w(z) = -1$.

The main bet of the cosmological community to explain the current accelerated expansion of the Universe is the Dark Energy (DE), a theoretical conception whose nature is still unknown [42–44]. The standard cosmological model Lambda Cold Dark Matter (ΛCDM) showed above,assumes the DE being a cosmological constant, has had great achievements such as and excellent agreement with most of the currently available data, i.e. Cosmic Microwave Background [45], Supernovae IA (SNeIa) [46] and Baryon Acoustic Oscillations (BAO) [47]. Nevertheless, the ΛCDM model has its own drawbacks: on the theoretical side fine tuning and cosmic coincidence [48, 49], and from an observational point of view, it also suffers from the Hubble tension [50], amongst others. These issues open the door to many other cosmological models, either by considering a dynamical DE, or by modifying the general theory of relativity [51].

With the intention of better explaining the nature of Dark Energy several surveys [45, 52, 53] have been carried out to obtain a greater amount of cosmological observations such as SNeIa, BAO, galaxies, large scale structure [54] and Cosmic Microwave Background (CMB). Nowadays, cosmological models present great challenges because, to be validated, they must match with all the relevant observational data.

A step further to the standard model is to consider the dark energy being dynamic, where the evolution of its EoS is usually parameterised. A commonly used form of $w(z)$ is to take into account the next contribution of a Taylor expansion in terms of the scale factor $w(a) = w_0 + (1-a)w_a$ or in terms of redshift $w(z) = w_0 + \frac{z}{1+z}w_a$ (CPL model: [55, 56]). The parameters $w_0$ and $w_a$ are real numbers such that at the present epoch $w|_{z=0} = w_0$ and $dw/dz|_{z=0} = -w_a$, and we recover ΛCDM when $w_0 = -1$ and $w_a = 0$. Hence the Friedmann equation for the CPL parameterisation turns out to be:

$$H(z)^2 = H_0^2\left[\Omega_{m,0}(1+z)^3 + (1-\Omega_{m,0})(1+z)^{3(1+w_0+w_a)}e^{-\frac{3w_a z}{1+z}}\right]. \tag{3.36}$$

As part of some simple models that allow deviations from ΛCDM we also use the polynomial-CDM model (PolyCDM) [57], that can be thought as a parameterisation of the Hubble function [58]. This model has the following Friedmann equation:

$$H(z)^2 = H_0^2\left[\Omega_{m,0}(1+z)^3 + \Omega_{1,0}(1+z)^2 + \Omega_{2,0}(1+z)^1 + (1-\Omega_{m,0}-\Omega_{1,0}-\Omega_{2,0})\right], \tag{3.37}$$

where $\Omega_{1,0}$ and $\Omega_{2,0}$ are two additional parameters, which within the ΛCDM both of them remain absent. Nevertheless, in [59] $\Omega_{2,0}$ is interpreted as a 'missing matter' component introduced to allow a symmetry that relates the big bang to the future conformal singularity. We recover ΛCDM when $\Omega_{1,0} = 0$ and $\Omega_{2,0} = 0$.

## 3.2 Cosmological observations

In cosmology, there are astronomical objects that allow the direct or indirect measure of a cosmological distance, this last quantity also can be calculated given the Friedmann equation of a certain cosmological model and then compare observational datasets with theoretical predictions.

### 3.2.1 Cosmic chronometers

Cosmic chronometers (CC) are galaxies, that evolve slowly and allow direct measures of the Hubble parameter $H(z)$. They are also known as Observational Hubble Data (OHD) and provide a direct measurement of the Hubble parameter and this is its advantage over other data like SNe-Ia and BAO. The differential age (DA) method measures $H(z)$ between two passively evolving galaxies that have similar metallicities and separated by a small redshift interval:

$$H(z) \simeq -\frac{1}{1+z}\frac{\Delta z}{\Delta t} \tag{3.38}$$

In this last equation, to obtain the rate $\frac{\Delta z}{\Delta t}$ is necessary measure the difference in age for two red galaxies at different redshifts [60]. Precisely these red galaxies receipt the name of cosmic chronometers. Another method with Luminous Red Galaxies (LRG) as cosmic chronometers, assumes that stars in LRG are all formed at the same time [61].

The respective statistical $\chi^2$ also can be calculated to the following way:

$$\chi_H^2 = \sum_{i=1}^{N} \frac{(H_{i,th} - H_{i,obs})^2}{\sigma_i^2} \tag{3.39}$$

where $\sigma_i$ is the error of the data. And with this $\chi^2$ we can use a MCMC method to estimate the parameters.

Cosmic chronometers measurements have been collected along several years [28, 62–68], and now 31 points are available with redshifts between 0.09 and 1.965, along with their statistical errors.

Given a Friedmann equation from a cosmological model then a theoretical value for $H(z)$ can be obtained and compared directly with these measurements.

### 3.2.2 Standard rulers: BAO

Standard rulers are astronomical objects with *known length*. A first example of standard rulers are the baryon acoustic oscillations (BAO), wich are fluctuations in the density of the visible baryonic matter caused by the decouple between photons and baryons by recombination.

In theory, the BAO scale is well determined and measures approximately 150Mpc, hence the BAO are considered standard rulers. Then, we can consider the angular diameter distance as the ratio of the comoving size of the object over its angular size:

$$d_A = \frac{c}{1+z} \int_0^z \frac{dz}{H(z)} = \frac{\Delta\chi}{\Delta\theta} \tag{3.40}$$

Note that the angular size is related with the cosmological parameters via $H(z)$. In the equation 3.40, $\chi$ is the known (by theory) size of the object and $\Delta\theta$ can be measured.

### 3.2.3 Standard candles: Type Ia supernovae

Standard candles are a class of cosmic objects for which all members have the same absolute magnitude $M$ and, among other contributions, the standard candles allowed the establishment of the expansion rate of the Universe. In the following equation $\mu(z)$ is the distance module, $M$ the absolute magnitude and $m_b(z)$ the apparent magnitude in the B band.

$$\mu(z) = m_b(z) - M \tag{3.41}$$

The Type Ia are a special class of supernovae, since they all have similar mass involved in their explosions, then also their absolute magnitudes are similar. Therefore, Type Ia supernovae can be made into standard candles by applying small corrections on their light curves (on the colour and shape) and they are useful to research about the expansion of the Universe, hence, of Dark Energy.

We can say that type Ia supernovae (SNeIa) have a *known luminosity*, therefore we can know the luminosity distance, which is a measure of the integral of the Hubble factor $H(z)$, and this quantity provides information about the cosmological parameters of a certain model.

As mentioned above, the traditional cosmology analysis [39] minimizes the statistic $\chi^2$ that is related to the theoretical values (predicted distance module) with the observed distance module:

$$\chi^2 = (\mu_{obs} - \mu_C)^t C_\mu^{-1} (\mu_{obs} - \mu_C) \tag{3.42}$$

Note that the decrease in $\chi^2$ means closer proximity between the theoretical value and the observational value.

The observed values, are calculated by the Spectral Adaptative Lightcurve Template 2 (SALT2) [69], that is the most general distance estimator linear:

$$\mu_{obs} = m_B^* - M_B + \alpha x_1 - \beta c \tag{3.43}$$

where $m_B^*$ refers to the peak magnitude in the B band of the rest-frame. If $x_1 = c = 0$ , $\alpha$ is the stretch correction and $\beta$ is the color correction.

On the other hand, $\mu_C$ can be calculated given the cosmological parameters:

$$\mu_C = 5log(\frac{d_L}{10pc}) \tag{3.44}$$

$$d_L(z,\theta) = (1+z)\frac{c}{H_0}\int_0^z \frac{dz'}{E(z',\theta)} \tag{3.45}$$

Besides, the covariance matrix is:

$$C_{SALT2} = \begin{bmatrix} \sigma^2_{x_0} & \sigma_{x_0,x_1} & \sigma_{x_0,c} \\ \sigma_{x_0,x_1} & \sigma^2_{x_1} & \sigma_{x_1,c} \\ \sigma_{x_0,c} & \sigma_{x_1,c} & \sigma^2_{x_2} \end{bmatrix}$$

The covariance matrix with the equation 3.42 allow us to build $\chi^2$ and the likelihood function for the Bayesian inference. See the references [70–72] for more details.

Lastly, related with the distance modulus function and the SNeIa data, if we consider a spatially flat universe, we have the following relationship between the luminosity distance ($d_L$) and the comoving distance $D(z)$:

$$d_L(z) = \frac{1}{H_0}(1+z)D(z), \qquad \text{with} \qquad D(z) = \int \frac{dz}{E(z)}, \tag{3.46}$$

where $E(z) = H(z)/H_0$. Finally, we can define the distance modulus as:

$$\mu(z) = 5\log d_L(z) + 25. \tag{3.47}$$

The SNeIa dataset used in this work is JLA, a compilation of 740 Type Ia supernovae. It is available a binned version that consists in 31 data points with a covariance matrix $C_{jla} \in \mathbb{R}^{31\times 31}$ [46].

### 3.2.4 Growth factor $f_{\sigma_8}$ measurements

The growth rate measurement is usually referred to the product of $f\sigma_8(a)$ where $f(a) \equiv d\ln\delta(a)/d\ln a$ is the growth rate of cosmological perturbations given by the density contrast $\delta(a) \equiv \delta\rho/\rho$, being $\rho$ the energy density and $\sigma_8$ the normalisation of the power spectrum on scales within spheres of $8h^{-1}$Mpc [73]. Therefore, the observable quantity $f\sigma_8(a)$ [or equivalently $f\sigma_8(z)$] is obtained by solving numerically

$$f\sigma_8(a) = a\frac{\delta'(a)}{\delta(1)}\sigma_{8,0}. \tag{3.48}$$

The $f\sigma_8$ data are obtained through the peculiar velocities from Redshift Space Distortions (RSD) measurements [74] observed in redshift survey galaxies or by weak lensing [75], where the density perturbations of the galaxies are proportional to the perturbations of matter. An extended version of the Gold-2017 compilation is available at [76], with 22 independent measurements of $f\sigma_8(z)$ from redshift space distortion measurements from a variety of surveys (see references therein).

## 3.3 SimpleMC code

SimpleMC [1] is a pure-Python package to cosmological parameter estimation, particularly to Dark Energy models. It was developed by Dr. José Alberto Vázquez and Dr. Anže Slosar. This code provides to cosmological researchers datasets, theory and algorithms and it already have defined several likelihoods functions in order to an easy use of the observational data and the theoretical models.

All Bayesian analysis performed in this work was with `SimpleMC` code. Figure 3.1 shows the overall structure and the parts where I had the opportunity to contribute in the coding (wrappers for external libraries or implementing certain routines) are in red boxes and briefly described in Table 3.

[1] https://github.com/ja-vazquez/SimpleMC

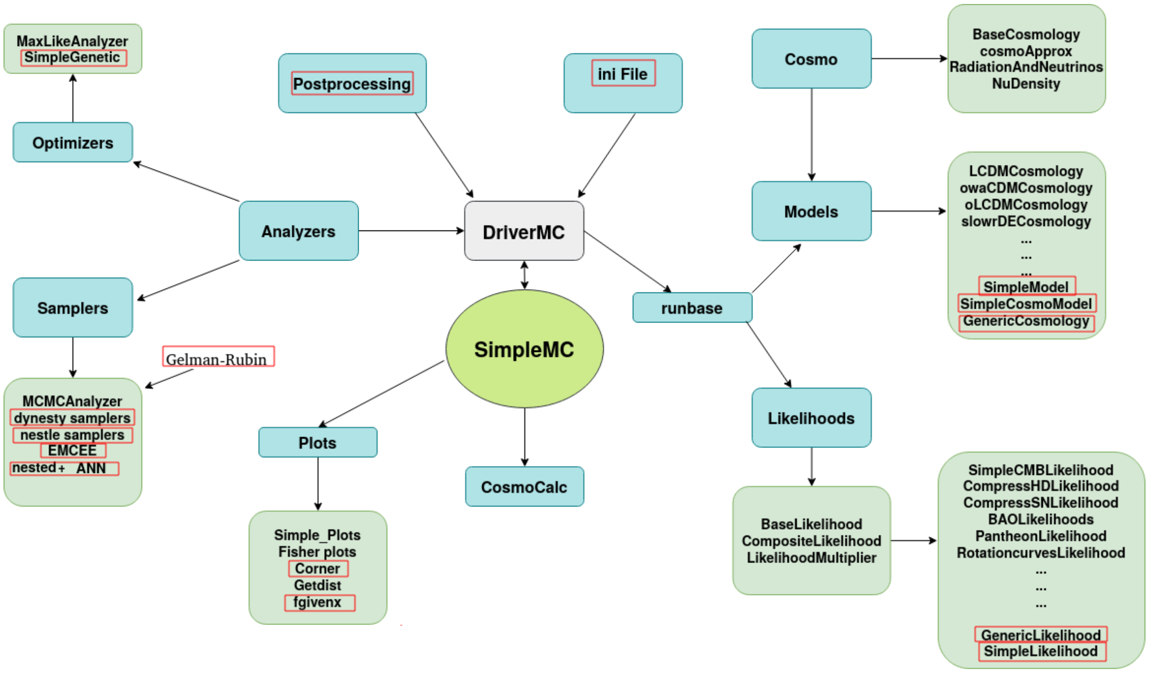


Figure 3.1: SimpleMC structure and contributions

| Contribution | Description |
|---|---|
| ini file | A configuration file to the end user. |
| Gelman-Rubin | Routine of Gelman-Rubin diagnostics to only one chain from Metropolis-Hasting algorithm. |
| Nested sampling | Wrapper for a modified version of dynesty library [77]. |
| Emcee | Wrapper to Emcee algorithm [78]. |
| Post-processing | Print and save output files and summaries. |
| Plots libraries | Wrappers for Getdist [79], corner [80] and fgivenx [81] libraries. |
| Generic classes | Python classes for generic and simple likelihoods and models. |
| Genetic algorithms | Simple genetic algorithm to optimization. |
| Neural networks | ANN to speed-up likelihood calculations. |

Table 3.1: Contributions in SimpleMC code

.

# CHAPTER 4

# SPEED-UP BAYESIAN INFERENCE

## 4.1 Introduction

Likelihood functions link the data to theory and are constructed by assuming some particular statistical distribution for the $D$ data, usually a Gaussian distribution. In the calculation of likelihood function, at a given point, it is also necessary to evaluate the theoretical model several times. On the other hand, if several types of observations are involved, the probability density function proposed as likelihood function should be a multiplication of several likelihoods (one for each type of data). The nature of Bayesian inference requires multiple evaluations of the likelihood function to generate a new sample with a higher likelihood value than its predecessor, and if these functions are complex, the computational time spent on these evaluations can be considerable.

In this chapter we evaluate the performance of an Artificial Neural Network (ANN) in the calculation of likelihood functions within a Bayesian inference process similar than [82, 83] and our tests were described in [2, 3]. In the following section we describe, in general terms, the procedure of our study and how we use the ANN. In Section 4.4, we show an application on the standard cosmological model and make the comparison of results of Bayesian inference using and not using the neural network to calculate the values of the likelihood function. Finally, the Section 4 contains our conclusions.

We based this work in the Ref. [82] that proposes a feedforward ANN to learn the likelihood function within a nested sampling algorithm [18]. As a first step, we use the pyBambi package [84], a python development based on [82], with MULTINEST [85] which is available in C language with a Python wrapper called `pymultinest`. We test the multinest with and without pyBambi with toys models to compare their samplings, it can be seen in Figure 4.1.

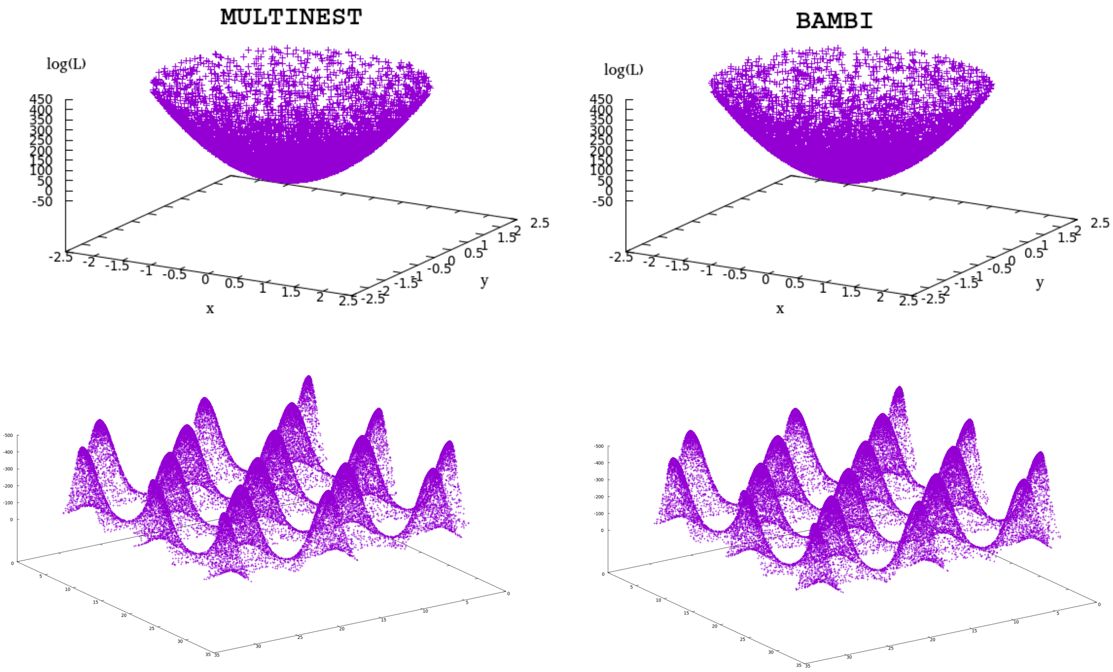


Figure 4.1: Comparison between sampling with MULTINEST and BAMBI

However, we wanted two different things. The first is a pure-Python way to implement a method of using the ANN with a nested algorithm, this allows an easy to install. Secondly, to use the speed-up method in cosmological parameter estimation. Therefore we use a nested sampling algorithm [85] available in Dynesty [77], we modify pyBambi [84] to our purposes and we implement our method in `SimpleMC`[1]. The ANN was implemented with the tensorflow Python library.

[1] www.github.com/ja-vazquez/SimpleMC

## 4.2 Artificial neural network architecture

The *Universal Approximation Theorem* [22] allows the use of an ANN to learn how to calculate the likelihood function. It states that an Artificial Neural Network with at least one hidden layer with a finite number of neurons can approach any continuous function if the activation function is continuous and non-linear.

In our case, we use a Feed-forward Neural Network with three hidden layers and the Rectified Linear Unit (RELU) as activation function. The loss function is the Mean Squared Error (MSE):

$$MSE = \frac{1}{n}\sum_{i}^{n}(y_i - \hat{y}_i)^2 \tag{4.1}$$

we apply the Adam gradient descent method to minimize it, initially with learning rate of 0.1 and reducing it by a factor of 0.1, until 0.0001, if through 5 epochs the value of the loss function does not improve.

The likelihood function evaluates points in the parameter space that have as many coordinates as free parameters have the theoretical model considered to make the Bayesian inference. As will be described later, the model used in this paper has three free parameters. Therefore, the number of nodes in the input layer of the neural network must match this value. On the other hand, the output layer have a single node that is the prediction of the likelihood function.

We tune the hyper-parameters of the neural network running 35 combinations of them and choosing the one that achieve a lower value for the loss function. For this test we use 50, 100, 150, 200, 250 and 300 nodes for the three hidden layers and 4, 8, 16, 32 and 64 for the batch size value. The combinations of hyperparameters are represented in Figure 4.2 and the best of them was the shown in Figure 4.3 with 8 for the batch size.

## 4.3 Method

In the generation of every 500 new samples, within the Bayesian inference framework, the neural network was trained. Therefore the 80% of this 500 samples with their respective likelihoods are used as training set, the remaining samples conform the test set.

When the MSE is below a predefined value (in our case, we use 0.1), the trained neural network replaces the analytical calculations of the likelihood function, otherwise the ANN is retrained after another 500 samples are generated and in the meantime the analytical

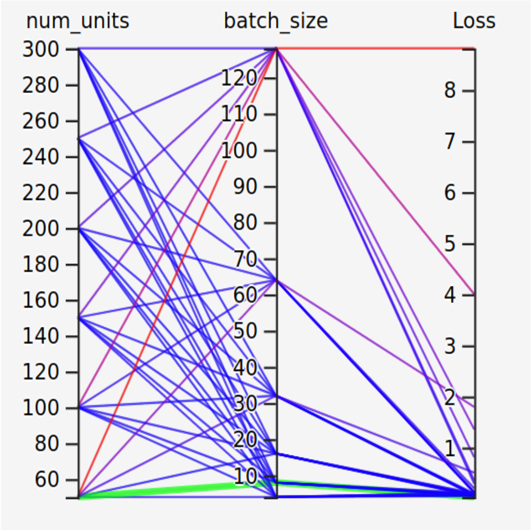


Figure 4.2: Hyperparameter grid to ANN that learn likelihood of ΛCDM with CC+BAO+JLA.

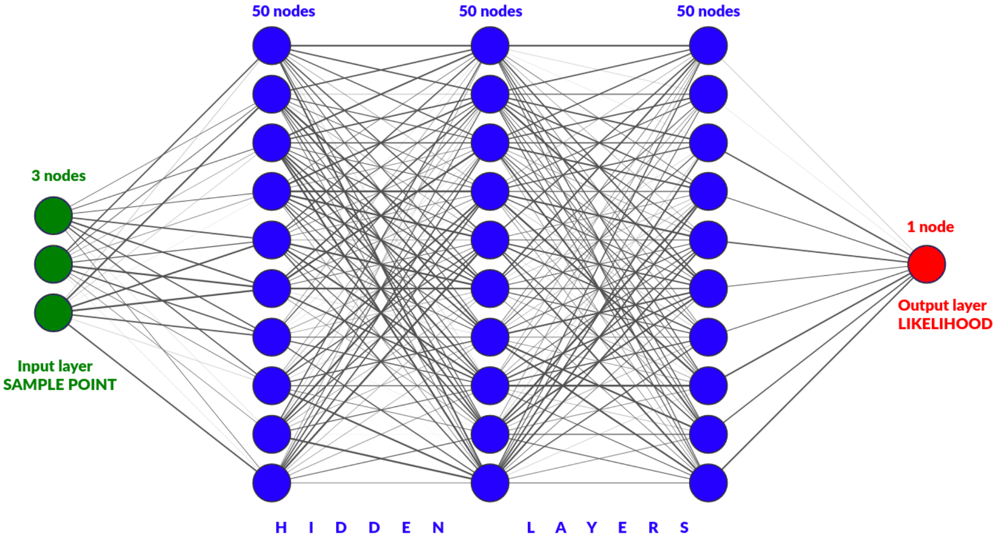


Figure 4.3: Feedforward ANN with three hidden layers used to learn likelihood function of ΛCDM with CC+BAO+JLA datasets.

likelihood continues to be used.

If, using the neural network, its predictions are outside the range of existing likelihoods (with a small deviation as tolerance, 0.1 in our case), the neural network is no longer used and the analytical calculation is returned. Subsequently, if the neural network is retrained correctly in the way described above, it can come back into action. In a few steps, we do the following:

1. Select a number of samples for the training set and validation set.
2. Choose a value for the loss function in which the neural network is considered to predict probabilities.
3. Define a criterion to evaluate whether the value of the likelihood function generated by the ANN is good or not. We force the value of the likelihood predicted to be between the low and high value of the last training set.

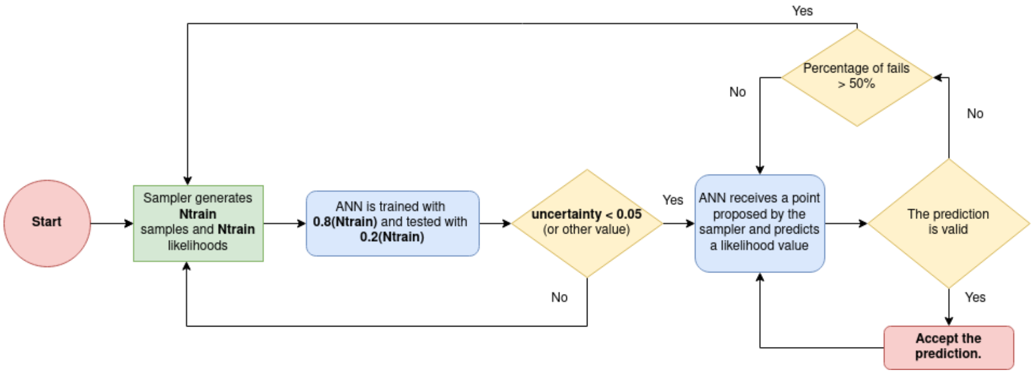

Yes
Percentage of fails > 50%
No
No
Start
Sampler generates **Ntrain** samples and **Ntrain** likelihoods
ANN is trained with **0.8(Ntrain)** and tested with **0.2(Ntrain)**
**uncertainty < 0.05** (or other value)
Yes
ANN receives a point proposed by the sampler and predicts a likelihood value
The prediction is valid
No
Yes
**Accept the prediction.**

Figure 4.4: Flow of the method

## 4.4 Bayesian inference on the ΛCDModel

The standard cosmological model, also known as ΛCDM, represents a flat universe with a cosmological constant that provides accelerated expansion. We use the Friedmann's equation, with a constriction for the cosmological constant energy density $\Omega_\Lambda = 1 - \Omega_m$ ($\Omega_m$ is the matter energy density), to reduce the number of free parameters:

$$\frac{H^2(a)}{H_0^2} = \frac{1}{h^2}\frac{\Omega_{0b}h^2 + \Omega_{0c}h^2}{a^3} + (1 - \Omega_{0m}), \tag{4.2}$$

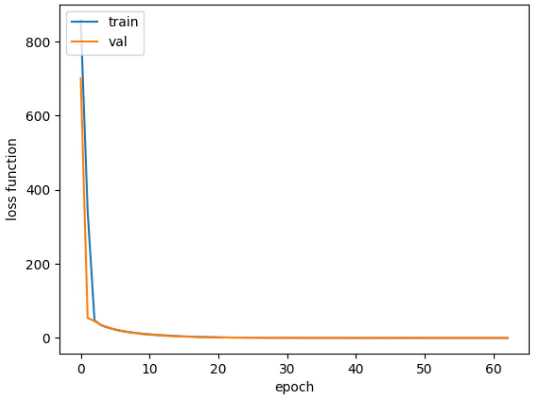


Figure 4.5: Loss function of the neural network in the training and validation set

where $H$ is the Hubble factor, $H_0$ the Hubble constant, $a$ is the scale factor (function of time representing the relative expansion of the universe), $\Omega_{0b}$ is the current energy density of baryons, $\Omega_{0c}$ the current energy density of cold dark matter, $\Omega_{0m} = \Omega_{0b} + \Omega_{0c}$ and $h = \frac{H}{100}$. Therefore, we use three free parameters: $h^2$, $\Omega_{0m}$ and $\Omega_{0b}h^2$. There are well known in cosmology [86] and allow us to evaluate our results of the Bayesian inference with and without neural network.

If we assume a Gaussian distribution for the data, we can construct the log-likelihood function as a chi-square test involving the theoretical model of the Equation 4.2 and the observational data. In our test, the likelihood function considers data from Type-Ia Supernovae [46], Cosmic Chronometers [87], Baryon Acoustic Oscillations and a compressed information of Planck-15 [57].

The figure 4.5 shows the behavior of the loss function (MSE) for the ANN, described in the previous section, in the training and validation sets, using 500 samples of the Bayesian inference process. If the value for MSE is high, it is very likely that the predictions made by the ANN will be wrong, so it is necessary to wait until the final stage of sampling in order to properly use the neural network predictions. Figure 4.6 shows the 1D and 2D plots of the resulting posterior distribution for Bayesian inference with and without ANN, we implemented this in the SimpleMC repository [88].

The parameter estimation process for ΛCDM was performed five times, both with and without ANN. The averages of these five procedures for each case are reported in table 4.1. We can note that the parameter estimation by nested sampling with and without the neural

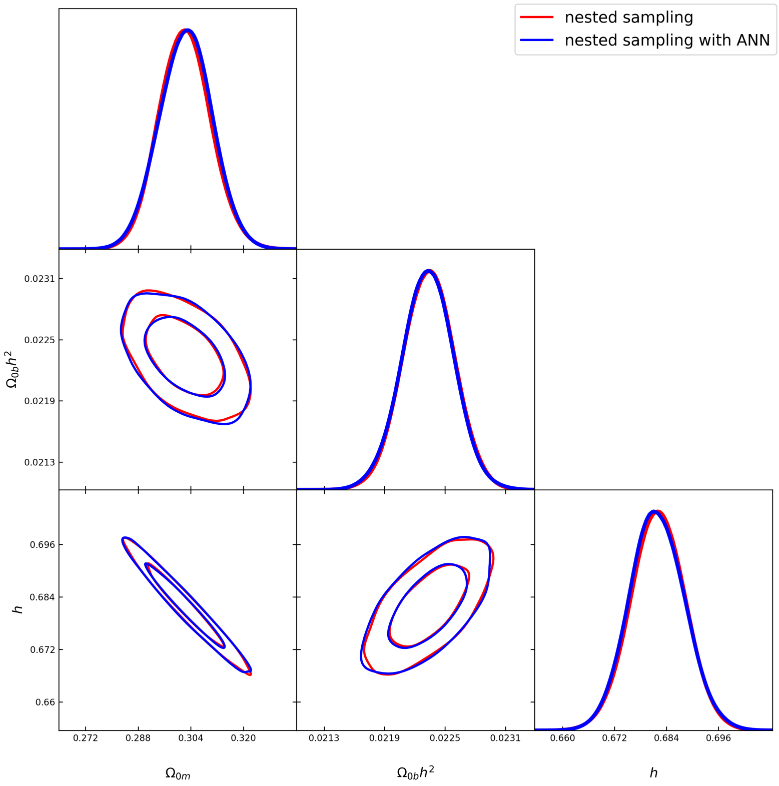


Figure 4.6: Posterior samples for the ΛCDM model obtained by nested sampling with and without neural network.

network are very close to each other and can be statistically interpreted in the same way. To get closer to the reference values, it would be necessary to add even more data to our Bayesian inference and that is not the purpose of this work.

We found that, on average, the neural network only calculates about 6% of the total likelihood calls and generates 4% of the total samples. However, in our example, this reduces the computational time by about 9.1 percent.

## 4.5 Conclusions

According to our results, for the standard cosmological model and the data sets mentioned above, we have noticed that if the neural network is well calibrated and achieves a low MSE, it can substitute the analytical calculation of the likelihood function in the final part

|  | Reference value [86] | without ANN | with ANN |
|---|---|---|---|
| $\Omega_m$ | $0.3166 \pm 0.0084$ | $0.2978 \pm 0.0680$ | $0.2982 \pm 0.0660$ |
| $\Omega_b h^2$ | $0.02242 \pm 0.00014$ | $0.0224 \pm 0.0009$ | $0.0224 \pm 0.0010$ |
| $h$ | $0.6727 \pm 0.006$ | $0.6918 \pm 0.0734$ | $0.6906 \pm 0.0723$ |
| $log(Bayesian\ evidence)$ |  | $-41.890 \pm 0.196$ | $-41.849 \pm 0.195$ |
| Samples generated with dynesty |  | 7742 | 7700 |
| Samples generated with ANN predictions |  |  | 282 |
| Likelihood predicted with ANN |  |  | 2202 |
| Total likelihood calls |  | 33007 | 33323 |
| time (minutes) |  | 73.2 | 66.8 |

Table 4.1: Results of the nested sampling algorithm applied to ΛCDM model

of a Bayesian inference process without significant alterations in the statistical analysis. Although the samples generated with the neural network likelihood predictions make up a small percentage, the acceleration in the Bayesian inference process is noticeable.

As future work we want to test this technique with more complex models, both in the field of cosmology and in any other branch of science in which a Bayesian inference process can be applied. We are also interested in testing with larger data sets and implementing it in parallel.

# CHAPTER 5

# NON-PARAMETRIC COSMOLOGICAL RECONSTRUCTIONS

## 5.1 Introduction

In this Chapter we show the main results of [5], in which we use cosmological datasets from cosmic chronometers, $f_{\sigma 8}$ measurements and SNeIa to reconstruct with Artificial Neural Networks, since a non-parametric inference, the observable cosmological functions provided by these data. In addition, we make a cosmological analysis of the results of our method and we compare it with the theoretical prediction from ΛCDM, CPL and PolyCDM models.

As part of a consistency test of our results, we generate synthetic data with the models generated by the trained neural networks and then, we perform Bayesian inference with `SimpleMC`. We compare the posterior PDF sampling using synthetic data with the respective result using the orgininal observational datasets.

## 5.2 Neural networks calibration

Throughout this work we use FFNNs for datasets with diagonal covariance matrices and autoencoders when correlations between the measurements are present, for instance within the JLA dataset.

In general, for the three types of cosmological observations (CC, $f\sigma_8$ and SNeIa)

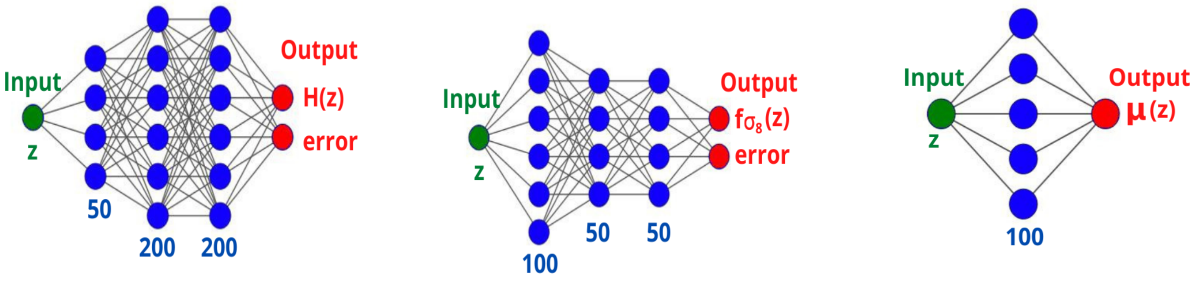


Figure 5.1: Neural network architectures chosen for cosmic chronometers (CC), $f\sigma_8$ measurements and distance modulus in JLA respectively. In the same order, the batch size found for each case was: 16, 1 and 1. The blue numbers indicate the nodes in each layer. In the last architecture, there is only one node in the output layer because the errors are calculated with a variational autoencoder (described in the Chapter 6) given the original covariance matrix of the systematic errors.

we have followed the next steps to find out a suitable neural network model for the corresponding data:

- We train several neural network configurations to gain insights about the complexity that their architecture require to model the data. According to the results of the loss function, we choose a number of layers.

- Several values are suggested for each hyperparameter of the neural network, based on the intuition achieved in the first step, a grid is formed that must be traversed to find the combination that provides the minimum value of the loss function. Among the hyperparameters it is the batch size, the number of nodes per layer and, in some cases, the dropout.

- The best ANN architectures found for each case are shown in Figure 5.1. The first two correspond to the CC and $f\sigma_8$ datasets respectively, for which 320 combinations were tested up to three fixed hidden layers: number of nodes in $\{50,100,150,200\}$ and the batch size in $\{1,4,8,16,32\}$. We found that for the compressed JLA dataset a one-layer neural network works best, so we refined the third architecture among 20 combinations, varying the number of nodes in $\{30,50,100,150,200\}$ and the batch size in $\{1,2,4,8\}$.

- We train the neural network with the combination of hyperparameters chosen in the previous step with a correct number of epochs. We verify the behaviour of the loss function in the training and validation sets to check that our model is neither underfitted nor overfitted. The effect of the epochs in the learning process using the first two ANN architectures of Figure 5.1 is shown in Figure 5.2.

- Once the neural network is trained, we can generate synthetic data points with absent redshift in the original datasets.
- By making several predictions with the neural network, the reconstruction of the data can be appreciated and compared with the original data. If the statistical behaviour of the synthetic data is not consistent, the neural networks must be retrained.
- We store the output, for a certain number of predicted data points as well as the new covariance matrix in order to be able to do the Bayesian inference of the cosmological models with these artificial data.
- We compare the parameter estimation of the synthetic data with the original set to verify they are statistically consistent and to analyse their differences. For this purpose, we use the SimpleMC[1] package [88], initially released at [57], along with a modified version of the *dynesty* nested sampling library [77], which allows to do the parameter estimation and Bayesian evidence calculation.

We have developed a package called CRANN[2] that contains the ANNs models already trained to produce synthetic cosmological data given a set of arbitrary redshifts. All the ANNs used in this work and their hyperparameter tuning were based on Tensorflow[3] and Keras[4] Python libraries.

In the case of cosmic chronometers and $f\sigma_8$ measurements we use FFNNs. These types of data have a diagonal covariance matrix and hence it can be arranged into a single column of the same length as the number of measurements. Therefore, these two datasets have three features (columns): the redshift $z$, the function $f(z)$, and the related error (taken from the diagonal of the covariance matrix). Once the FFNN is trained, for a given of set of points (redshifts) we can output the cosmological function together with its simulated statistical error. It should be noted that the neural networks learn to generate the cosmological function and the error, where the latter is the result of modelling the original statistical errors from the observational data set.

[1] www.github.com/ja-vazquez/SimpleMC

[2] Cosmological Reconstructions with Artificial Neural Networks (CRANN). https://github.com/igomezv/crann

[3] www.tensorflow.org

[4] www.keras.io

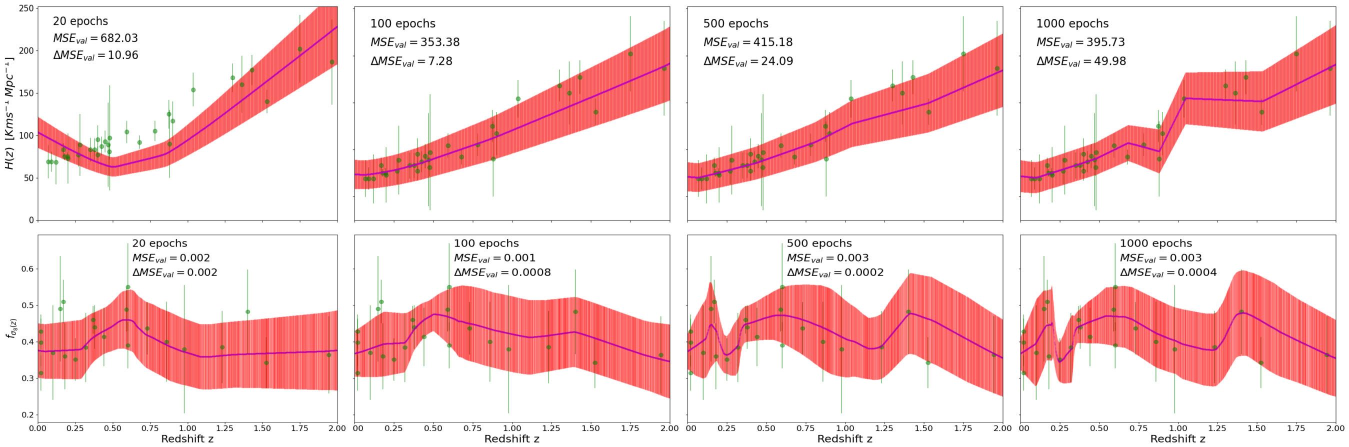


Figure 5.2: Effect of the number of epochs in the training with the CC dataset (top) and with the $f\sigma_8(z)$ measurements (bottom). The first case (20 epochs) shows underfitting, while considering 1000 epochs shows overfitting. In the $f\sigma_8$ dataset, the cases for 500 and 1000 epochs present overfitting. In both cases, we choose 100 epochs due to the lower values of MSE and ΔMSE in the validation set. Green points display real data-points with error bars, and in purple synthetic data along with red error bars.

On the other hand, through the analysis of the JLA SNeIa compilation, we also use a FFNN to learn the behaviour of distance modulus $\mu(z)$ in a similar fashion we did for the CC and $f\sigma_8$. However, in order to handle the full covariance matrix we use a VAE as described in the Chapter 6; using this type of neural network allows us to map the distribution of the distance modulus to the distribution of the coded representation of the autoencoder to generate new covariance matrices. One restriction of this method to bear in mind is that the new matrix has to have the same dimension as the original one. However, we can generate a matrix given any combination of new redshifts, provided that this set has the same length as the original measurements.

In addition to the above procedure, we slightly modify the FFNNs to implement an epistemic calculation of their uncertainties using the dropout [37]. In this way, we add dropout between the layers of the FFNNs and run the Monte Carlo dropout several times to obtain average values and uncertainties for each prediction. We combine our FFNN designs with the implementation of MC-DO layers from `astronn`[5][89], and compare the results of this method with the previous ANNs implementations. Because dropout is a regularisation technique, the number of epochs is irrelevant for a large enough set. The error predictions

[5] https://github.com/henrysky/astroNN

and the uncertainties are independent, therefore the total standard deviation is:

$$\sigma = \sqrt{er_p + \sum_i u_i^2}, \tag{5.1}$$

where $u_i$ is the epistemic uncertainty involved with the FFNN used and $er_p$ is the error prediction.

As mentioned above, once the non-parametric reconstruction is obtained we can generate synthetic data and use them to perform a Bayesian inference procedure. For these purposes, we use the following flat priors: for the matter density parameter today $\Omega_m \in [0.05,\ 0.5]$, for the physical baryon density parameter $\Omega_b h^2 \in [0.02,\ 0.025]$, for the reduced Hubble constant $h \in [0.4,\ 0.9]$, and for the amplitude of the (linear) power spectrum $\sigma_8 \in [0.6,\ 1.0]$. When assuming the CPL parameterisation, we use $w_0 \in [-2.0,\ 0.0]$ and $w_a \in [-2.0,\ 2.0]$; and for the PolyCDM model, we use $\Omega_1 \in [-1.0,\ 3]$ and $\Omega_2 \in [-0.5,\ 3]$. The $h$ parameter refers to the dimensionless reduced Hubble parameter $H/100\text{kms}^{-1}\text{Mpc}^{-1}$.

## 5.3 Results

In order to perform the reconstructions of the Hubble parameter $H(z)$, the growth rate measurement $f\sigma_8(z)$ and the distance modulus $\mu(z)$ we apply two different methods by implementing the feedforward neural networks shown in Figure 5.1: i) using the trained neural network (FFNN) and ii) along with the FFNN, by considering uncertainties with the Monte Carlo dropout (FFNN+MC-DO). To test the quality of our ANNs predictions we perform the Bayesian inference procedure for the CPL and PolyCDM models with the original data and with the data generated by the neural networks. Then, and to improve the constraints on the free parameters, in some cases we also include a compressed version of Planck-15 information (treated as a BAO experiment located at redshift $z = 1090$ [57]).

### Reconstruction of $H(z)$

Once the chosen FFNN is trained with the CC dataset (green points in Figure 5.3), we input new redshift values ($z$) and let the network to predict the corresponding values for $H(z)$ and their errors. In the left-panel of Figure 5.3, we generate 1000 synthetic data (magenta points) with their respective error bars (red lines) with the FFNN. These errors are the result of the ANN modelling the errors contained in the original dataset.

Besides the intrinsic error associated with the datasets, we consider an uncertainty related with the FFNN by adding a Monte Carlo dropout between each layer of the chosen FNNN architecture, under the method described in [37]. Among several tests to dropout values between $[0, 0.5]$, we found that a good value for the dropout is 0.3 and we trained the FFNN with MC-DO along 1000 epochs. After training the new FFNN with dropout between each layer, we can make the predictions to 1000 unseen redshifts and, with 100 executions of MC-DO for each prediction. We obtain the right-panel of Figure 5.3, that contains the total standard deviation considering the uncertainties of the neural network. In this case, the result is not continuous due to the variations caused by the probabilistic nature of the MC-DO and, now, the error bars contain information about the statistical uncertainty of the FFNN trained model; indeed the error is larger than in the FFNN alone case.

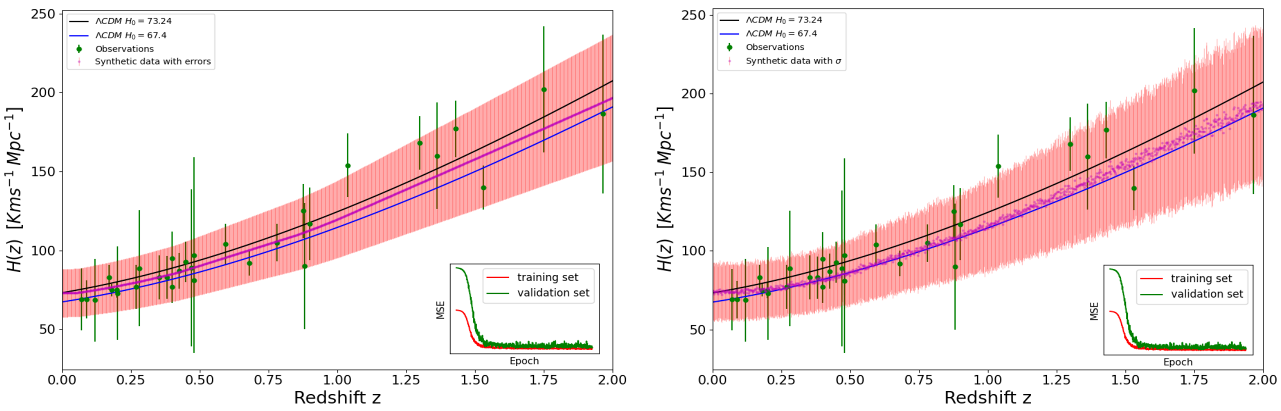


Figure 5.3: $H(z)$ reconstructions produced with 1000 synthetic data points generated with FFNNs. *Left*: Purple line represents the FFNN predictions for $H(z)$ along with their error bars in red colour. *Right*: Similarly to FFNN but adding MC-DO, we executed 100 times the Monte Carlo dropout to compute the uncertainties of the predictions, therefore the purple points are the average predictions of the MC-DO executions and the red error bars are the uncertainties of the FFNN plus the error predictions (see Equation 5.1). In both cases, we compare the non-parametric reconstruction with the original cosmic chronometers (green bars) and $H(z)$ from ΛCDM, as shown in the labels. The small panels displayed the receptive behaviour of the loss function (MSE) in the training (red) and validation (green) sets.

An interesting feature shown in both panels of Figure 5.3 is that, despite the original dataset does not contain a measurement for $H(z = 0)$, the FFNN prediction is $H_{pred}(z =$

$0) = 75.09 \pm 15.49$ km s$^{-1}$ Mpc$^{-1}$ and the prediction of the FFNN with Monte Carlo dropout is $H_{pred}(z=0) = 77.07 \pm 15.91$ km s$^{-1}$ Mpc$^{-1}$. Both results have a better matching to the one measured with Cepheid variables [87]: $H_0 = 73.24$ km s$^{-1}$ Mpc$^{-1}$. In addition, it can be appreciated that as the redshift increases, the model generated by the FFNNs approaches ΛCDM with $H_0 = 67.40$ km s$^{-1}$ Mpc$^{-1}$ measured by Planck mission [45]. It is worth to remember that the ANN models do not have any prior cosmological or statistical assumption so they have been built solely from the data.

## Reconstruction of $f\sigma_8(z)$

Similarly to the Hubble parameter case, we apply the same methodology but now with measurements of the $f\sigma_8$ function [76]. In order to generate the reconstruction plot shown in the left panel of Figure 5.4, once the FFNN is trained, we generate 1000 synthetic data points (red points). Besides, we put the evaluation of $f\sigma_8(z)$ from CPL model in three different scenarios of $w_0$ and $w_a$.

We added Monte Carlo dropout to the FFNN, shown in the second ANN architecture of Figure 5.1, to be able to calculate the uncertainties of the ANN. We train this new FFNN along 2000 epochs. In this case, among several tests to dropout values between $[0, 0.5]$, we choice a dropout of 0.1 because it had the best performance. Then we obtain, with 1000 synthetic $f\sigma_8$ data points, the reconstruction of the right-panel of Figure 5.4, where the purple line is the average obtained by MC-DO predictions and the error bars contain an error conformed by the standard deviations (uncertainties) of MC-DO for each prediction plus the error predictions. We can notice, that in both cases, the models plotted are within the reconstruction and hence this dataset by itself may provide loose constraints on the CPL parameters. However, the values $w_0 = -0.8$ and $w_a = -0.4$ (brown line) seem to have a better agreement with the reconstruction, as we shall see below.

## Distance modulus $\mu(z)$ reconstructions

Regarding to the distance modulus $\mu(z)$, we train a FFNN (last ANN in Figure 5.1) for a given redshift. We assume a gaussian distribution for the predictions of the distance modulus and using a trained VAE we produce a new covariance matrix (see Chapter 6). Once the FFNN is trained, we can generate synthetic data points for unseen redshifts and reconstruct the $\mu(z)$ function as it can be appreciate in the left panel of Figure 5.5. In

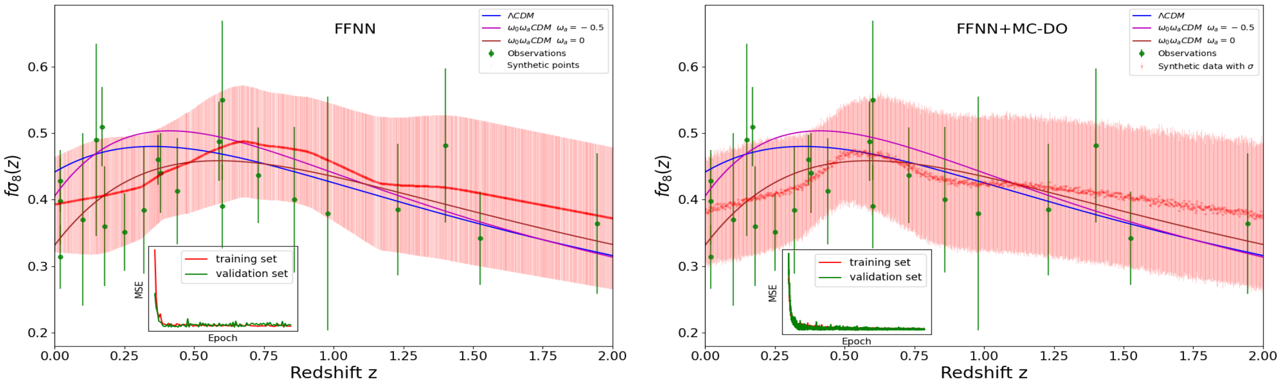


Figure 5.4: Reconstruction of $f\sigma_8(z)$ with 1000 synthetic data points (red dots) and their respective errors (red bars) learned by ANNs. *Left*: FFNN alone, red line is conformed by the predictions of $f\sigma_8$. *Right*: FFNN using Monte Carlo dropout, the averages of 100 executions of MC-DO are indicated with the red line and their standard deviations are added to the error predictions. In both cases the small panels displayed the receptive behaviour of the loss function (MSE) in the training (red curve) and validation (green curve) sets.

particular, we use 31 log-uniformly distributed redshifts over the interval $z \in [0.01, 1.3]$. To apply MC-DO, we did several tests to dropout values in the interval $[0, 0.5]$ and we found that a dropout with 0.01 value has a good performance. We executed 100 times MC-DO to obtain the right panel of Figure 5.5.

An important point to bear in mind is that when using a full covariance matrix we need to restrict to 31 synthetic data points in order to generate the covariance matrix with the VAE for mapping to the new points in the latent space. See the next chapter for details.

## Parameter estimation with synthetic data

Neural networks allow us to produce data models with several parameters (neural network weights) which are uninterpretable, however with the use of synthetic data generated by these models we can analyze them with Bayesian inference and compare their results with those obtained from the original data. Thus we performed the Bayesian inference analysis to estimate the best fit parameters of the CPL and PolyCDM models. The aim of this procedure is to look for possible deviations of the ΛCDM model with the neural networks approach.

In addition to the three original datasets (cosmic chronometers, $f\sigma_8$ measurements and binned JLA compilation), we have created two datasets for each type of observation from the trained FFNNs with and without MC-DO. As a proof of the concept, the new datasets

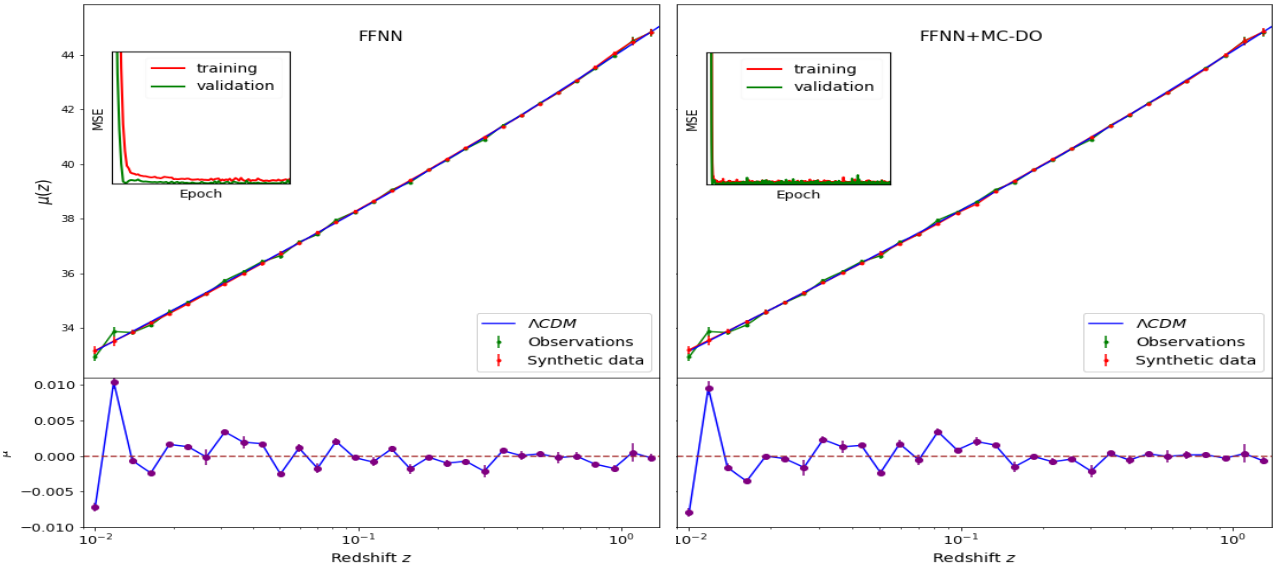


Figure 5.5: *Left*: 31 new data points (red dots) generated with FFNN. *Right*: 31 new data points (red dots) generated with FFNN+MC-DO. In both cases it is shown their receptive behaviour of the loss function (MSE) in the training (red curve) and validation (green curve) sets along the chosen number of epochs for each case (300 and 1800). The 31 green dots are the original points from the binned version of JLA.

for CC and $f\sigma_8$ consisted of 50 random uniformly distributed points in redshift, while for SNeIa they were 31 log-uniformly distributed in redshift (same size as the original dataset). For the SNeIa case we also generated its respective covariance matrix with the decoder part of the trained VAE.

We have used the data from CC, $f\sigma_8$ measurements and JLA separately, and also some combinations of them. The most representative results are in Figure 5.6 along with Tables 5.1 and 5.2, which contains mean values and standard deviations, and they have been sorted according to the datasets used as a source (original, FFNN, and FFNN+MC-DO), and to the models involved (ΛCDM, CPL, and PolyCDM). It is indicated when the Planck point has been added to the data sets. Results are displayed for the reduced Hubble parameter $h$, $\sigma_8$, $w_0$ and $w_a$ measurements for the CPL model, and $\Omega_1$ with $\Omega_2$ when the model is PolyCDM. In addition, the last column of the Table 5.1 and Table 5.2 contains the $-2\log\mathscr{L}_{max}$ of the Bayesian inference process for each case. Before analysing each case separately, it is worth mentioning that there are some generalities in the results. In general, it can be noted that when using a single source separately (no Planck information added) the constraints are consistent among each other, that is, they all have a similar best-fit (maximum likelihood) and are in agreement with the $\Lambda$CDM model.

In the case of parameter estimation using exclusively the CC dataset, the first two panels of Figure 5.6 (and first block in Table 5.1) show that the best-fit values are mutually contained within their $1\sigma$ standard deviations and are in agreement with the $\Lambda$CDM values. However, we can notice that when Planck information is added the reduced Hubble parameter value slightly increases for the ANNs. For instance, for the CPL model with original data the constraints are $h = 0.673 \pm 0.046$ whereas for the synthetic data it increases to $0.713 \pm 0.059$ for the FFNN and $0.726 \pm 0.063$ for the FFNN+MC-DO; in fact these values obtained by the synthetic data are closer to the Hubble parameter value of the Cepheid variables than to the Planck mission value. This issue, as a supplement to Figure 5.3, shows that the neural network models generated by cosmic chronometers are sensing the Hubble tension, although considering the size of the standard deviation values, all the results of the parameter estimation are still statistically consistent with each other. Something similar happens when added Planck information to JLA SNeIa and assuming the CPL model. With the original data the constraints are $h = 0.695 \pm 0.021$ whereas for the synthetic data they increase to $0.704 \pm 0.025$ for the FFNN and $0.712 \pm 0.026$ for the FFNN+MC-DO. However both datasets are still statistically consistent, within $1\sigma$, with the $\Lambda$CDM parameters. If taken into account the JLA+CC combination with the Planck information, the increment of the reduced Hubble parameter is still present but also a small deviation of $\Lambda$CDM (about $1\sigma$) for FFNN+MC-DO, with constraints of $w_0 = -0.957 \pm 0.141$ and $w_a = -0.563 \pm 0.669$.

On the other hand, considering only measurements of $f\sigma_8$+Planck for the synthetic data, the $w_0$ and $w_a$ constraints suggest a slight deviation from $\Lambda$CDM. With the FFNN data the values are $w_0 = -0.657 \pm 0.172$ and $w_a = -0.493 \pm 0.265$, and for FFNN+MC-DO we have $w_0 = -0.673 \pm 0.183$ and $w_a = -0.364 \pm 0.221$. In fact, it can be seen in the Figure 5.6 that the cosmological constant is right on the limits of the $2\sigma$ contours.

By using all datasets combined CC+$f\sigma_8$+JLA we have performed a Bayesian inference to the three models $\Lambda$CDM, CPL and PolyCDM. Compared to the original datasets we found consistency throughout the models with the $\Lambda$CDM parameters and slight shift when using the synthetic data, for instance higher values for the reduced hubble parameter, lower for the $\sigma_8$ parameter and for $w_0$: $-0.916 \pm 0.065$ (FFNN) and $-0.925 \pm 0.068$ (FFNN+MC-DO). Deviations of the standard values are enhanced once we use synthetic data along with Planck information. This can be seen on the constraints of the PolyCDM model for the

FFNN source: $\Omega_1 = 0.272 \pm 0.194$, $\Omega_2 = -0.092 \pm 0.058$. Also, based on the improvement in the fit alone ($\sqrt{2\Delta \log \mathscr{L}_{max}}$), and using the same source, we found a preference to the data for the CPL model of $1.5\sigma$ and $1.7\sigma$ for the PolyCDM. That is, the Artificial Neural Network by itself is finding deviations from the standard cosmological model.

The above discussion suggests that, if the models generated by the neural networks are correct, hypothetical new observations within the range of the existing ones would tend to move away from ΛCDM. Therefore, parameter estimation in the CPL and PolyCDM models in conjunction to the models generated by the neural networks suggest that ΛCDM does not have the best match to the data. In all cases, the addition of the Planck point increases the tension and the need for a model beyond ΛCDM.

To reinforce the idea that models generated by the neural networks depart from ΛCDM, from the posterior distribution samples for CPL, we obtained the posterior distribution of its corresponding EoS, as shown in Figure 5.7 (using `fgivenx` Python library [81]). From these plots it can be seen that $w = -1$ (value corresponding to ΛCDM) lies in the most probable region within $1\sigma$ with the original data; however, in the case of the synthetic data the cosmological constant moves away from the most probable region, still within $1\sigma$ without considering the Planck point, and outside $1\sigma$ when it is taken into account.

Finally, as part of the Bayesian analysis, we can perform a model comparison with Bayesian evidences $Z$ through the Bayes' factor $B$ and the Jeffrey's scale [90]. Table 5.3 shows the log-Bayes' factor of ΛCDM compared to CPL and PolyCDM models using the different sources of data. It can be seen that with the synthetic data the penalisation of having extra parameters decreases from strong advantage to an inconclusive advantage due to the improvement of the fit in both models. However, it is worth noting that ΛCDM stays with a slight advantage.

| Source | CPL | PolyCDM |
|---|---|---|
| Original | 3.651 | 2.837 |
| FFNN | 1.823 | 0.687 |
| FFNN+MC-DO | 2.464 | 1.159 |

Table 5.3: Log-Bayes' factor $\ln(B) = \ln(Z_{\Lambda CDM}) - \ln(Z)$ of ΛCDM with respect the other models using the same data source for each case. The combined dataset used in this table is JLA+CC+$f_{\sigma 8}$+Planck

.

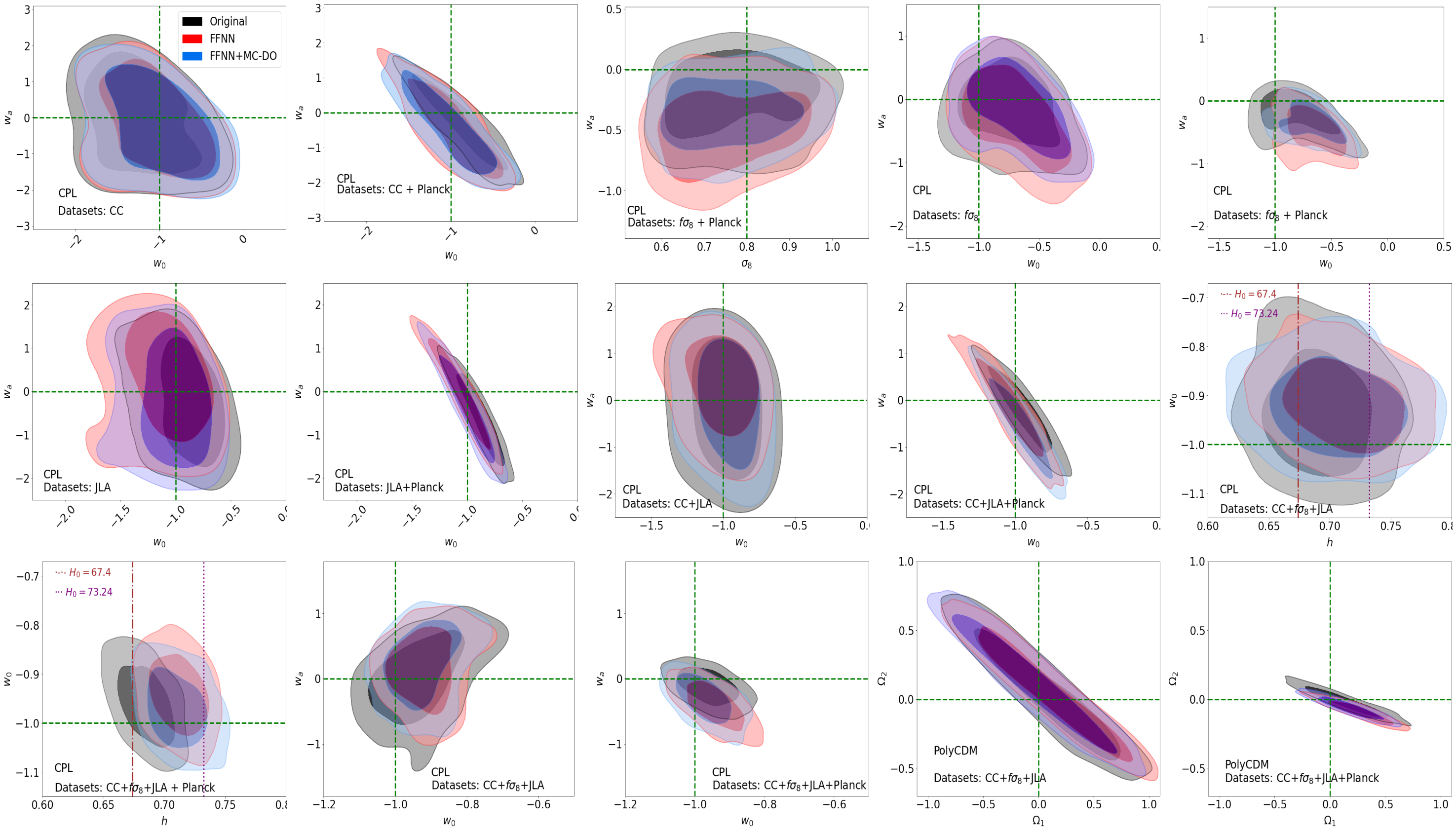


Figure 5.6: 2D marginalised posterior distributions from different combinations of datasets: original data, synthetic datasets from FFNN and FFNN+MC-DO. The green dashed lines ($w_0 = -1$, $w_a = 0$) and ($\Omega_1 = 0$, $\Omega_2 = 0$) correspond to the $\Lambda$CDM model. The constraints are plotted with $1\sigma$ and $2\sigma$ confidence contours.

## 5.4 Conclusions

Throughout this work, we have shown that well-calibrated artificial neural networks have the capacity to produce non-parametric reconstructions from which synthetic cosmological data, statistically consistent with the originals, can be generated even when the datasets are small.

We have explored the generation of synthetic covariance matrices through VAE, and the results have allowed us to carry out Bayesian inference without drawbacks. However, for larger datasets, we believe that it will be convenient to use convolutional layers in the autoencoder and a slightly different approach to dealing with the computing demand.

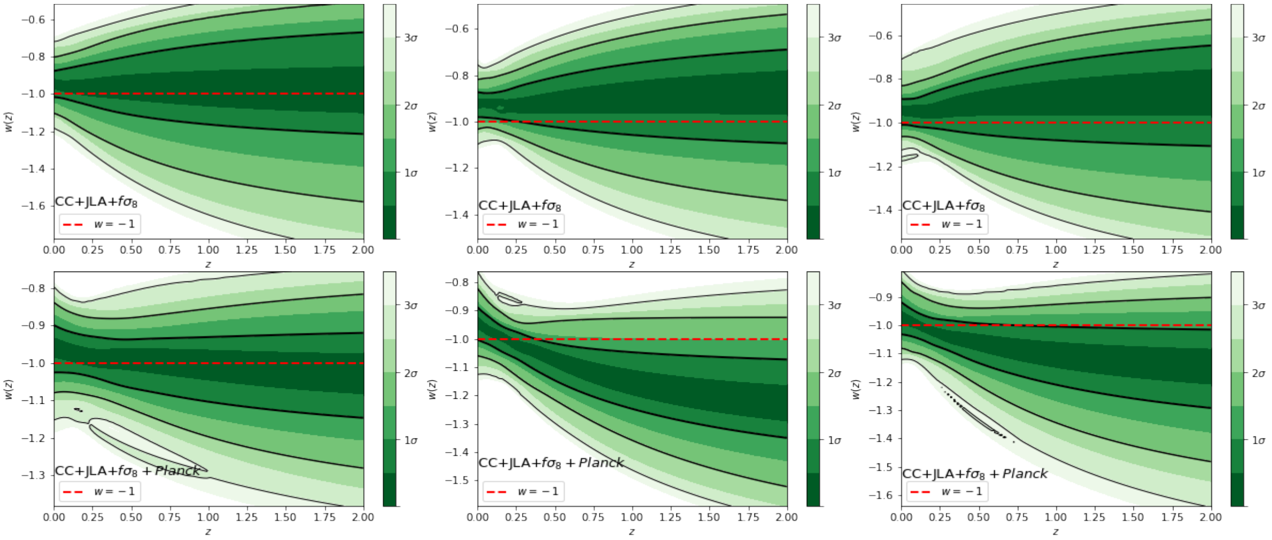


Figure 5.7: Posterior probability distribution functions of the Dark Energy EoS considering CPL parameterisation, by using original data, FFNN and FFNN+MC-DO respectively. Planck-15 point information is additionally included in the lower panels. $1-3\sigma$ confidence intervals are plotted as black lines.

Using the Monte Carlo dropout method allows to have more information about the predictions of neural networks through their epistemic estimation of uncertainty. The results obtained have also contributed to both the methodological and the cosmological analysis and validate the outputs from the ANNs without this method as they are very similar.

The models generated by the neural networks were produced exclusively from the data, therefore they offer the possibility of reconstructing cosmological functions without assumptions about the data distribution and without assuming any cosmological model as a starting point. From the non-parametric reconstructions produced with the neural networks, we were able to observe how the Hubble parameter changes as the cosmic chronometers are at higher redshifts, as suggested by the current Hubble tension. We could also note that SNeIa are observations very much in accordance with ΛCDM and, in contrast, that ΛCDM is not the best model to describe $f\sigma_8$ measurements. Overall, using Bayesian inference on the CPL and PolyCDM models with the synthetic neural network data, we have observed that the ΛCDM model does not perfectly match these data and loses some of the advantage given by the original observations.

It is worth mentioning that the cosmological results obtained in this work are limited to the current observational used and have been sufficient to show some interesting cosmological features from the data alone. We have shown that our method can be a good complement to the traditional Bayesian analysis and, moreover, could be applied to other types of cosmological observations and models. In this way, we can see that the use of neural networks, from their models created for the data and the generation of synthetic data, can complement the analysis of cosmological models and improve the interpretations of their behaviours. We plan to apply similar techniques to other types of cosmological data, including the complete set of covariance matrices.

| Source | Model | | Datasets: | **CC** | | | $-2\ln\mathcal{L}_{max}$ |
|---|---|---|---|---|---|---|---|
| | | | $h$ | $w_0$ | $w_a$ | | |
| Original | ΛCDM | - | $0.678 \pm 0.039$ | $--$ | $--$ | | 14.502 |
| | CPL | - | $0.703 \pm 0.064$ | $-1.223 \pm 0.447$ | $-0.061 \pm 1.075$ | | 14.290 |
| | CPL | + Planck | $0.673 \pm 0.046$ | $-0.867 \pm 0.326$ | $-0.325 \pm 0.824$ | | 14.638 |
| FFNN | ΛCDM | - | $0.698 \pm 0.057$ | $--$ | $--$ | | 0.176 |
| | CPL | - | $0.703 \pm 0.071$ | $-1.072 \pm 0.431$ | $-0.179 \pm 1.025$ | | 0.042 |
| | CPL | + Planck | $0.713 \pm 0.059$ | $-0.962 \pm 0.337$ | $-0.485 \pm 0.890$ | | 0.120 |
| FFNN+MC-DO | ΛCDM | - | $0.699 \pm 0.063$ | $--$ | $--$ | | 0.346 |
| | CPL | - | $0.689 \pm 0.078$ | $-1.014 \pm 0.450$ | $-0.227 \pm 1.003$ | | 0.284 |
| | CPL | + Planck | $0.726 \pm 0.063$ | $-1.029 \pm 0.355$ | $-0.377 \pm 0.897$ | | 0.808 |
| | | | Datasets: | $f\sigma_8$ | | | $-2\ln\mathcal{L}_{max}$ |
| | | | $h$ | $w_0$ | $w_a$ | $\sigma_8$ | |
| Original | ΛCDM | - | $0.648 \pm 0.147$ | $--$ | $--$ | $0.787 \pm 0.115$ | 11.932 |
| | CPL | - | $0.638 \pm 0.135$ | $-0.742 \pm 0.264$ | $-0.144 \pm 0.468$ | $0.777 \pm 0.111$ | 11.908 |
| | CPL | + Planck | $0.648 \pm 0.062$ | $-0.801 \pm 0.229$ | $-0.225 \pm 0.254$ | $0.771 \pm 0.109$ | 11.944 |
| FFNN | ΛCDM | - | $0.650 \pm 0.144$ | $-$ | $--$ | $0.694 \pm 0.172$ | 0.292 |
| | CPL | - | $0.648 \pm 0.142$ | $-0.701 \pm 0.271$ | $-0.290 \pm 0.540$ | $0.777 \pm 0.111$ | 0.284 |
| | CPL | + Planck | $0.628 \pm 0.046$ | $-0.657 \pm 0.172$ | $-0.493 \pm 0.265$ | $0.756 \pm 0.109$ | 0.374 |
| FFNN+MC-DO | ΛCDM | - | $0.651 \pm 0.147$ | $--$ | $--$ | $0.652 \pm 0.170$ | 0.984 |
| | CPL | - | $0.632 \pm 0.140$ | $-0.674 \pm 0.270$ | $-0.156 \pm 0.489$ | $0.775 \pm 0.110$ | 0.960 |
| | CPL | + Planck | $0.622 \pm 0.046$ | $-0.673 \pm 0.183$ | $-0.364 \pm 0.221$ | $0.756 \pm 0.103$ | 1.038 |
| | | | Datasets: | **JLA** | | | $-2\ln\mathcal{L}_{max}$ |
| | | | $h$ | $w_0$ | $w_a$ | | |
| Original | ΛCDM | - | $0.638 \pm 0.146$ | $--$ | $--$ | | 33.214 |
| | CPL | - | $0.652 \pm 0.141$ | $-0.901 \pm 0.238$ | $-0.216 \pm 0.899$ | | 32.354 |
| | CPL | + Planck | $0.695 \pm 0.021$ | $-0.880 \pm 0.140$ | $-0.606 \pm 0.696$ | | 30.528 |
| FFNN | ΛCDM | - | $0.645 \pm 0.144$ | $--$ | $--$ | | 14.670 |
| | CPL | - | $0.640 \pm 0.137$ | $-1.092 \pm 0.277$ | $0.287 \pm 0.957$ | | 13.888 |
| | CPL | + Planck | $0.704 \pm 0.025$ | $-1.061 \pm 0.178$ | $-0.018 \pm 0.811$ | | 14.808 |
| FFNN+MC-DO | ΛCDM | - | $0.643 \pm 0.142$ | $--$ | $--$ | | 16.446 |
| | CPL | - | $0.641 \pm 0.135$ | $-1.037 \pm 0.248$ | $-0.245 \pm 0.996$ | | 16.274 |
| | CPL | + Planck | $0.712 \pm 0.026$ | $-0.994 \pm 0.165$ | $-0.395 \pm 0.802$ | | 16.504 |

Table 5.1: Parameter estimation using Bayesian inference with datasets from different sources: original, FFNN alone and FFNN using Monte Carlo dropout.

| | | | Datasets: | **CC+JLA** | | | $-2\ln\mathscr{L}_{max}$ |
|---|---|---|---|---|---|---|---|
| | | | $h$ | $w_0$ | $w_a$ | | |
| Original | ΛCDM | - | $0.690\pm0.030$ | $--$ | $--$ | | 47.822 |
| | CPL | - | $0.687\pm0.030$ | $-0.980\pm0.173$ | $-0.156\pm0.939$ | | 47.830 |
| | CPL | + Planck | $0.687\pm0.018$ | $-0.946\pm0.137$ | $-0.232\pm0.592$ | | 47.918 |
| FFNN | ΛCDM | - | $0.705\pm0.037$ | $--$ | $--$ | | 14.846 |
| | CPL | - | $0.695\pm0.037$ | $-1.010\pm0.165$ | $0.315\pm0.715$ | | 14.096 |
| | CPL | + Planck | $0.708\pm0.021$ | $-1.031\pm0.150$ | $-0.167\pm0.688$ | | 15.478 |
| FFNN+MC-DO | ΛCDM | - | $0.703\pm0.038$ | $--$ | $--$ | | 16.808 |
| | CPL | - | $0.698\pm0.039$ | $-0.968\pm0.155$ | $-0.046\pm0.859$ | | 16.688 |
| | CPL | + Planck | $0.717\pm0.019$ | $-0.957\pm0.141$ | $-0.563\pm0.669$ | | 17.252 |
| | | | Datasets: | **CC+JLA+** $f\sigma_8$ | | | $-2\ln\mathscr{L}_{max}$ |
| | | | $h$ | $w_0$ | $w_a$ | $\sigma_8$ | |
| Original | ΛCDM | - | $0.695\pm0.032$ | $--$ | $--$ | $0.795\pm0.115$ | 60.244 |
| | ΛCDM | +Planck | $0.690\pm0.013$ | $--$ | $--$ | $0.790\pm0.109$ | 60.33 |
| | CPL | - | $0.692\pm0.029$ | $-0.933\pm0.086$ | $0.009\pm0.476$ | $0.763\pm0.111$ | 59.840 |
| | CPL | + Planck | $0.685\pm0.015$ | $-0.961\pm0.057$ | $-0.122\pm0.194$ | $0.763\pm0.112$ | 59.832 |
| FFNN | ΛCDM | - | $0.721\pm0.034$ | $--$ | $--$ | $0.786\pm0.117$ | 16.278 |
| | ΛCDM | +Planck | $0.704\pm0.013$ | $--$ | $--$ | $0.7500\pm0.105$ | 19.191 |
| | CPL | - | $0.712\pm0.032$ | $-0.916\pm0.065$ | $0.150\pm0.432$ | $0.786\pm0.114$ | 15.076 |
| | CPL | + Planck | $0.712\pm0.015$ | $-0.941\pm0.057$ | $-0.417\pm0.246$ | $0.733\pm0.100$ | 17.044 |
| FFNN+MC-DO | ΛCDM | - | $0.713\pm0.035$ | $--$ | $--$ | $0.775\pm0.116$ | 18.096 |
| | ΛCDM | +Planck | $0.702\pm0.012$ | $--$ | $--$ | $0.753\pm0.100$ | 20.842 |
| | CPL | - | $0.706\pm0.037$ | $-0.925\pm0.068$ | $0.222\pm0.443$ | $0.763\pm0.105$ | 17.734 |
| | CPL | + Planck | $0.711\pm0.017$ | $-0.970\pm0.055$ | $-0.318\pm0.247$ | $0.723\pm0.097$ | 19.034 |
| | | | | $\Omega_1$ | $\Omega_2$ | | $-2\ln\mathscr{L}_{max}$ |
| Original | PolyCDM | - | $0.693\pm0.029$ | $0.089\pm0.416$ | $0.034\pm0.302$ | $0.788\pm0.108$ | 59.832 |
| | PolyCDM | +Planck | $0.696\pm0.017$ | $0.147\pm0.238$ | $-0.020\pm0.071$ | $0.779\pm0.107$ | 59.972 |
| FFNN | PolyCDM | - | $0.712\pm0.035$ | $0.110\pm0.444$ | $0.048\pm0.302$ | $0.776\pm0.109$ | 15.186 |
| | PolyCDM | +Planck | $0.736\pm0.021$ | $0.272\pm0.194$ | $-0.092\pm0.058$ | $0.781\pm0.105$ | 16.422 |
| FFNN+MC-DO | PolyCDM | - | $0.707\pm0.036$ | $-0.054\pm0.435$ | $0.134\pm0.302$ | $0.775\pm0.114$ | 17.730 |
| | PolyCDM | +Planck | $0.732\pm0.019$ | $0.173\pm0.195$ | $-0.062\pm0.057$ | $0.762\pm0.105$ | 18.936 |

Table 5.2: Parameter estimation using Bayesian inference with combinations of datasets from different sources: original, FFNN alone and FFNN using Monte Carlo dropout.

# CHAPTER 6

# RECONSTRUCTION OF COVARIANCE MATRIX

## 6.1 Introduction

This chapter is devoted to the treatment, commented in the Chapter 5, to the reconstruction of the covariance matrix of a SNeIa compilation dataset and is a further explanation on the appendix of [5].

Variational autoencoders are widely used in image processing and our developed method is a first approach to use VAE for numerical purposes and, in particular, to reconstruct covariance matrices, so further mathematical formality is a pending task.

A covariance matrix has certain similarities with an image file, such as both are two-dimensional matrices and the neighborhood of points has correlations. In an image, this neighborhood manifests itself in shapes or colors, i.e., for objects to be recognizable, between two neighboring points the color or shape does not change drastically. On the other hand, in covariance matrices the relationship between two points is effectively the covariance or correlation of the measurements; moreover, they also follow some patterns, for example, they are symmetric and often the elements of the diagonal (the variance terms) have higher values than the rest. Because of these similarities, we have proposed that a neural network, in particular a variational autoencoder (VAE), models a covariance matrix. We use the binned version of the JLA compilation of SNeIa that has a covariance matrix $C_{jla} \in \mathbb{R}^{31\times 31}$ related to both statistical and systematic measurement errors [46].

## 6.2 Methodology

Our first problem is that, in order to feed a neural network, it is necessary to have a dataset of covariance matrices, which is difficult when the covariance matrix includes both systematic and statistical errors. In general, for a given measurement dataset, there is only one covariance matrix, and these matrices usually include correlations of various nature, such as experimental calibration. So, is it possible to generate a dataset to train a neural network that can generate a new covariance matrix for the interpolated measurement values? Our intention has just been to propose a way to answer this question.

To have a dataset to train our VAE, we generate thousands of matrices by adding Gaussian noise of the same order of magnitude for each entry of the original covariance matrix.

The left panel of Figure 6.1 shows the chosen architecture for the VAE, where $\mu$ and $\sigma$ represent two layers that lie between the last encoder layer and the latent space; in this case both layers have a single neuron (the same dimension as the latent space). We have used a batch size of 32 and the hyperbolic tangent as the activation function. For practicality, since we are interested in mapping the distribution of the distance modulus to the latent space, we designed the VAE with a 1-dimensional latent space, so its mean $\mu$ and variance $\sigma$ are also 1-dimensional.

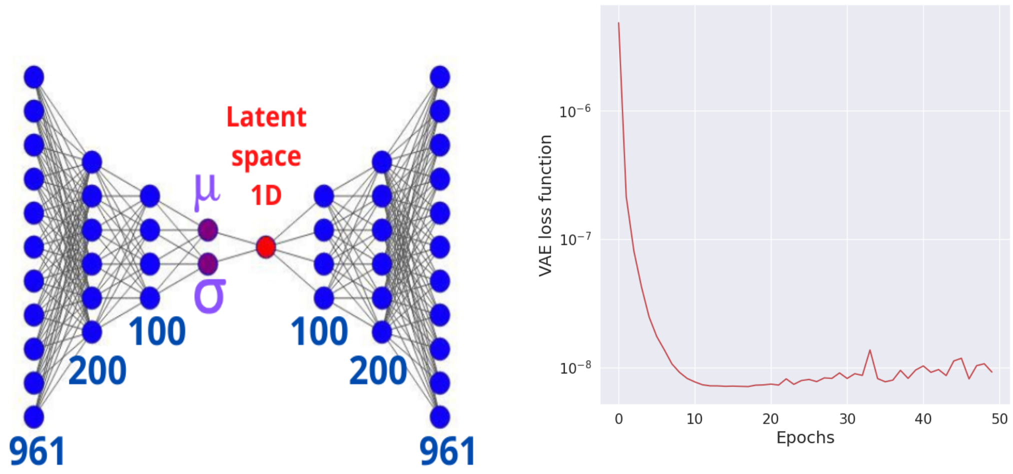


Figure 6.1: VAE architecture designed to generate synthetic covariance matrices from a point in the latent space.

With the created dataset of covariance matrices we train the autoencoder over 1000 epochs; the loss function plot (see Eq. 2.28) is shown in the right panel of Figure 6.1.

Once the VAE is trained, we can use the decoder part to generate new covariance matrices that traverse the latent space. In addition, we can explore the mean and variance layers by stepping through the entire set of training covariance matrices (Figure 6.2) and appreciate the distribution of the sampled latent space by variational inference with the sigma and mean layers (Figure 6.3).

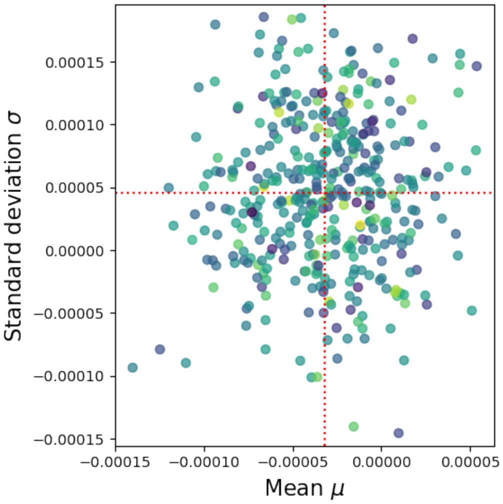


Figure 6.2: Samples of the mean and variance layers.

To generate covariance matrices from the predictions of the modular distances coming from the FFNNs, using their means and standard deviations we have assigned them a Gaussian distribution (Figure 6.4). We have related the original measurements to the most likely region of the latent space, and the deviations from the original measurements can be linearly mapped to the latent space to generate a new covariance matrix as shown in Figure 6.5. Once the VAE is trained, each element of the training set generates a value in the mean layer and in the variance layer.

It is worth commenting that if the predicted distance modulus distribution has a larger deviation from the original measurements, then the VAE sampling method must be adapted using a larger standard deviation. To illustrate this issue, in the following Python code for this method, the argument `stddev` should be incremented as required.

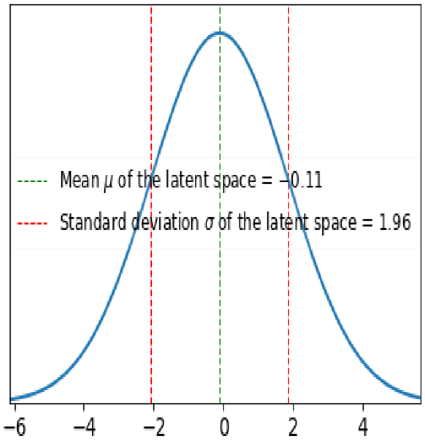


Figure 6.3: Sampled distribution of the latent space.

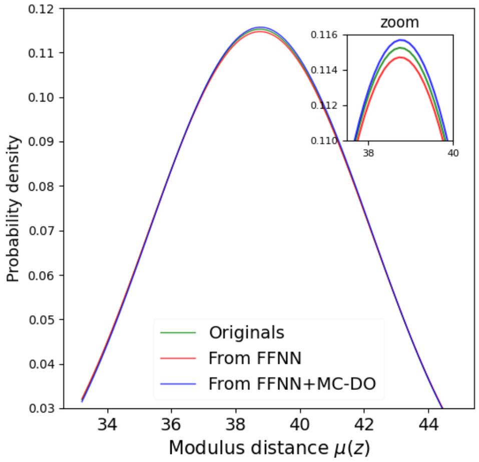


Figure 6.4: Comparison between the distributions assigned for the modular distances from different sources, these distributions are mapped into the latent space to generate a new covariance matrix with the VAE decoder.

```
def sampling(args):
    z_mean, z_log_sigma = args
    epsilon = K.random_normal(shape=(K.shape(z_mean)[0],
                              latent_dim),
                              mean=0.0, stddev=0.05)
    return z_mean + K.exp(z_log_sigma) * epsilon
```

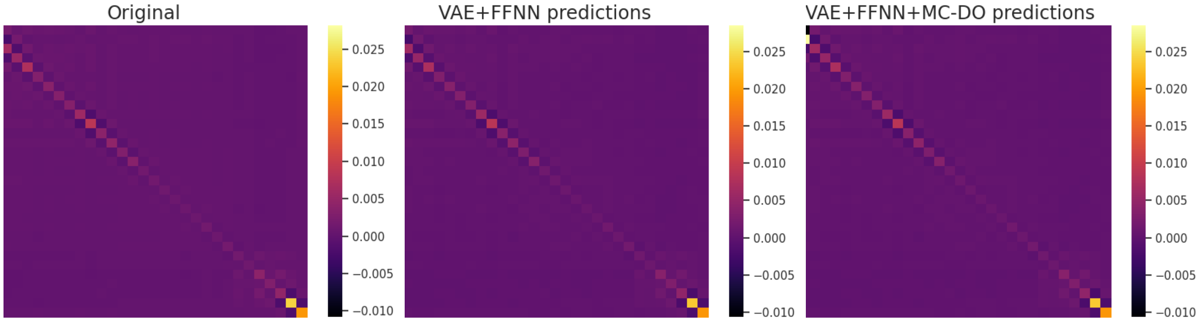


Figure 6.5: *Left*: Original covariance matrix with systematic errors from JLA compilation (binned version). Covariance matrices predicted by the VAE (*Middle*) and VAE with MC-DO (*Right*).

In summary, with the modulus distance predictions from the FFNN trained with the JLA SNeIa compilation (previous chapter), we compute the mean and standard deviation and assume a Gaussian distribution to map it into latent space. Then, using the decoder part of the VAE we obtain a new covariance matrix (an similar output to that in Figure 6.5).

## 6.3 Results

The developed method could be used to reconstruct covariance matrices of any origin. In our particular case, when using it for cosmological observations from supernovae, the parameter estimation shown in Chapter 6 shows that these matrices were consistent with the original ones and that they did not have any drawbacks. It is worth mentioning that at all times, before performing the full Bayesian analysis of the previous chapter, we had to test several synthetic matrices with Bayesian inference using few data and simple models to check their consistency.

The covariance matrix used in this work consisted of $31 \times 31 = 961$ elements and the linear arrangement of the VAE layers were effective. However, we could observe that for higher dimensional covariance matrices it would be necessary to employ convolutional layers and thus also dabble in the use of convolutional neural networks for numerical computations. In fact, this is a work on which we are already working.

# CHAPTER 7

## FINAL COMMENTS

With the work developed in this thesis, we have noticed that neural networks obtain very good results in non-linear modeling. Both in the calculation of likelihood and in the modeling of observational data sets. However, a crucial part is training and validation of results, especially when the available data sets are not very large.

We have observed that by correctly calibrating neural networks and not using them as "black boxes", a lot of useful numerical and statistical information can be extracted. In this way, its incorporation into the traditional Bayesian inference analysis is very useful. Furthermore, the non-parametric inference obtained with neural networks is also quite competent.

Regarding the cosmological part, we have discovered the great capacity that neural networks can offer for data analysis. With the analysis in Chapter 5, some of the current tensions in cosmology were observed without the need to involve any theoretical models beforehand. In the future we want to apply these techniques with different cosmological models to complement other types of analysis that have been carried out so far.

We have a lot of work to do to apply this type of method to more complex data or larger datasets. Also, the study of covariance matrices under this scheme is just beginning. In general, we believe that, for the data analysis methods discussed in this thesis, there are good research opportunities in the too near future.

# REFERENCES

[1] John Skilling et al. Nested sampling for general bayesian computation. *Bayesian analysis*, 1(4):833–859, 2006.

[2] Isidro Gómez-Vargas, Ricardo Medel Esquivel, Ricardo García-Salcedo, and J Alberto Vázquez. Neural network within a bayesian inference framework. *Journal of Physics: Conference Series*, 1723(1):012022, 2021.

[3] I Gómez-Vargas, R Medel-Esquivel, R García-Salcedo, and J A Vázquez. Una aplicación de las redes neuronales artificiales en la cosmología. *Komputer Sapiens*, 2(11):12–17, 2019.

[4] J. A. Vázquez, R Medel Esquivel, and I. Gómez-Vargas. Cosmología observacional con redes neuronales artificiales. *Memorias de la XXVII Escuela de Verano en Física*, 27(1):89, 2021.

[5] Isidro Gómez-Vargas, J Alberto Vázquez, Ricardo Medel Esquivel, and Ricardo García-Salcedo. Cosmological reconstructions with artificial neural networks. 2021. [arXiv:2104.00595].

[6] John E Freund and Frank Jefferson Williams. *Dictionary/outline of basic statistics*. Courier Corporation, 1991.

[7] Larry Wasserman. *All of nonparametric statistics*. Springer Science & Business Media, 2006.

[8] David JC MacKay and David JC Mac Kay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[9] Udo Von Toussaint. Bayesian inference in physics. *Reviews of Modern Physics*, 83(3):943, 2011.

[10] Andrew R Liddle. Information criteria for astrophysical model selection. *Monthly Notices of the Royal Astronomical Society: Letters*, 377(1):L74–L78, 2007.

[11] Andrew R Liddle, Pia Mukherjee, and David Parkinson. Cosmological model selection. *arXiv preprint astro-ph/0608184*, 2006.

[12] Adrian E Raftery. Approximate bayes factors and accounting for model uncertainty in generalised linear models. *Biometrika*, 83(2):251–266, 1996.

[13] Robert E Kass and Adrian E Raftery. Bayes factors. *Journal of the american statistical association*, 90(430):773–795, 1995.

[14] J. Alberto Vazquez, M. Bridges, M.P. Hobson, and A.N. Lasenby. Reconstruction of the Dark Energy equation of state. *JCAP*, 09:020, 2012. [arXiv:1205.0847].

[15] Ricardo Medel Esquivel, Isidro Gómez-Vargas, J Alberto Vázquez, and Ricardo García Salcedo. An introduction to markov chain monte carlo. *Boletín de Estadística e Investigación Operativa*, 1(37):47–74, 2021.

[16] Farhan Feroz, Jonathan R Gair, Michael P Hobson, and Edward K Porter. Use of the multinest algorithm for gravitational wave data analysis. *Classical and Quantum Gravity*, 26(21):215003, 2009.

[17] WJ Handley, MP Hobson, and AN Lasenby. Polychord: nested sampling for cosmology. *Monthly Notices of the Royal Astronomical Society: Letters*, 450(1):L61–L65, 2015. [arXiv:1502.01856].

[18] John Skilling. Nested sampling. In *AIP Conference Proceedings*, volume 735, pages 395–405. American Institute of Physics, 2004.

[19] Josh Speagle and Kyle Barbary. dynesty: Dynamic nested sampling package. *Astrophysics Source Code Library*, 2018.

[20] Devinderjit Sivia and John Skilling. *Data analysis: a Bayesian tutorial*. OUP Oxford, 2006.

[21] Stuart Geman, Elie Bienenstock, and René Doursat. Neural networks and the bias/variance dilemma. *Neural computation*, 4(1):1–58, 1992.

[22] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural networks*, 3(5):551–560, 1990.

[23] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[24] Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012.

[25] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[26] Rémi Bardenet, Mátyás Brendel, Balázs Kégl, and Michele Sebag. Collaborative hyperparameter tuning. In *International conference on machine learning*, pages 199–207, 2013.

[27] Frank Hutter, Holger H Hoos, Kevin Leyton-Brown, and Thomas Stützle. Paramils: an automatic algorithm configuration framework. *Journal of Artificial Intelligence Research*, 36:267–306, 2009. [arXiv:1401.3492].

[28] Cong Zhang, Han Zhang, Shuo Yuan, Siqi Liu, Tong-Jie Zhang, and Yan-Chun Sun. Four new observational $h(z)$ data from luminous red galaxies in the sloan digital sky survey data release seven. *Research in Astronomy and Astrophysics*, 14(10):1221, 2014. [arXiv:1207.4541].

[29] Hugo Larochelle, Dumitru Erhan, Aaron Courville, James Bergstra, and Yoshua Bengio. An empirical evaluation of deep architectures on problems with many factors of variation. In *Proceedings of the 24th international conference on Machine learning*, pages 473–480, 2007.

[30] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[31] Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural networks*, 2(1):53–58, 1989.

[32] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. 2013. [arXiv:1312.6114].

[33] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR, 2014.

[34] Carl Doersch. Tutorial on variational autoencoders. 2016. [arXiv:1606.05908].

[35] Diederik P Kingma and Max Welling. An introduction to variational autoencoders. 2019. [arXiv:1906.02691].

[36] Daniel Ramos, Javier Franco-Pedroso, Alicia Lozano-Diez, and Joaquin Gonzalez-Rodriguez. Deconstructing cross-entropy for probabilistic binary classifiers. *Entropy*, 20(3):208, 2018.

[37] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Insights and applications. In *Deep Learning Workshop, ICML*, volume 1, page 2, 2015.

[38] David H Weinberg, Michael J Mortonson, Daniel J Eisenstein, Christopher Hirata, Adam G Riess, and Eduardo Rozo. Observational probes of cosmic acceleration. *Physics reports*, 530(2):87–255, 2013.

[39] Adam G Riess, Alexei V Filippenko, Peter Challis, Alejandro Clocchiatti, Alan Diercks, Peter M Garnavich, Ron L Gilliland, Craig J Hogan, Saurabh Jha, Robert P Kirshner, et al. Observational evidence from supernovae for an accelerating universe and a cosmological constant. *The Astronomical Journal*, 116(3):1009, 1998.

[40] Saul Perlmutter, G Aldering, M Della Valle, S Deustua, RS Ellis, S Fabbro, A Fruchter, G Goldhaber, DE Groom, IM Hook, et al. Discovery of a supernova explosion at half the age of the universe. *Nature*, 391(6662):51–54, 1998.

[41] David N Spergel, R Bean, O Doré, MR Nolta, CL Bennett, Joanna Dunkley, G Hinshaw, N ea Jarosik, E Komatsu, L Page, et al. Three-year wilkinson microwave anisotropy probe (wmap) observations: implications for cosmology. *The Astrophysical Journal Supplement Series*, 170(2):377, 2007.

[42] Edmund J Copeland, Mohammad Sami, and Shinji Tsujikawa. Dynamics of dark energy. *International Journal of Modern Physics D*, 15(11):1753–1935, 2006. [arXiv: hep-th/0603057].

[43] Luca Amendola and Shinji Tsujikawa. *Dark energy: theory and observations*. Cambridge University Press, 2010.

[44] Pilar Ruiz-Lapuente. *Dark energy: observational and theoretical approaches*. Cambridge University Press, 2010.

[45] Nabila Aghanim, Yashar Akrami, M Ashdown, J Aumont, C Baccigalupi, M Ballardini, AJ Banday, RB Barreiro, N Bartolo, S Basak, et al. Planck 2018 results-vi. cosmological parameters. *Astronomy & Astrophysics*, 641:A6, 2020. [arXiv:1807.06209].

[46] M et al Betoule, R Kessler, J Guy, J Mosher, D Hardin, R Biswas, P Astier, P El-Hage, M Konig, S Kuhlmann, et al. Improved cosmological constraints from a joint analysis of the sdss-ii and snls supernova samples. *Astronomy & Astrophysics*, 568:A22, 2014. [arXiv:1401.4064].

[47] Shadab Alam, Metin Ata, Stephen Bailey, Florian Beutler, Dmitry Bizyaev, Jonathan A Blazek, Adam S Bolton, Joel R Brownstein, Angela Burden, Chia-Hsun Chuang, et al. The clustering of galaxies in the completed sdss-iii baryon oscillation spectroscopic survey: cosmological analysis of the dr12 galaxy sample. *Monthly Notices of the Royal Astronomical Society*, 470(3):2617–2652, 2017. [arXiv:1607.03155].

[48] Varun Sahni. The cosmological constant problem and quintessence. *Classical and Quantum Gravity*, 19(13):3435, 2002. [arXiv: astro-ph/0202076].

[49] P James E Peebles and Bharat Ratra. The cosmological constant and dark energy. *Reviews of modern physics*, 75(2):559, 2003. [arXiv: astro-ph/0207347].

[50] Stephen M Feeney, Daniel J Mortlock, and Niccolo Dalmasso. Clarifying the hubble constant tension with a bayesian hierarchical model of the local distance ladder. *Monthly Notices of the Royal Astronomical Society*, 476(3):3861–3882, 2018. [arXiv:1707.00007].

[51] Austin Joyce, Lucas Lombriser, and Fabian Schmidt. Dark energy versus modified gravity. *Annual Review of Nuclear and Particle Science*, 66:95–122, 2016. [arXiv:1601.06133].

[52] Kevork N Abazajian, Jennifer K Adelman-McCarthy, Marcel A Agüeros, Sahar S Allam, Carlos Allende Prieto, Deokkeun An, Kurt SJ Anderson, Scott F Anderson, James Annis, Neta A Bahcall, et al. The seventh data release of the sloan digital sky survey. *The Astrophysical Journal Supplement Series*, 182(2):543, 2009. [arXiv:0812.0649].

[53] Michael E Levi, Lori E Allen, Anand Raichoor, Charles Baltay, Segev BenZvi, Florian Beutler, Adam Bolton, Francisco J Castander, Chia-Hsun Chuang, Andrew Cooper, et al. The dark energy spectroscopic instrument (desi). 2019. [arXiv:1907.10688].

[54] Volker Springel, Carlos S Frenk, and Simon DM White. The large-scale structure of the universe. *Nature*, 440(7088):1137–1144, 2006.

[55] Michel Chevallier and David Polarski. Accelerating universes with scaling dark matter. *International Journal of Modern Physics D*, 10(02):213–223, 2001. [arXiv: gr-qc/0009008].

[56] Eric V Linder. Exploring the expansion history of the universe. *Physical Review Letters*, 90(9):091301, 2003. [arXiv: astro-ph/0208512].

[57] Éric Aubourg, Stephen Bailey, Julian E Bautista, Florian Beutler, Vaishali Bhardwaj, Dmitry Bizyaev, Michael Blanton, Michael Blomqvist, Adam S Bolton, Jo Bovy, et al. Cosmological implications of baryon acoustic oscillation measurements. *Physical Review D*, 92(12):123516, 2015. [arXiv:1411.1074].

[58] Zhongxu Zhai, Michael Blanton, Anže Slosar, and Jeremy Tinker. An evaluation of cosmological models from the expansion and growth of structure measurements. *The Astrophysical Journal*, 850(2):183, 2017. [arXiv:1705.10031].

[59] J Alberto Vázquez, S Hee, MP Hobson, AN Lasenby, M Ibison, and M Bridges. Observational constraints on conformal time symmetry, missing matter and double dark energy. *Journal of Cosmology and Astroparticle Physics*, 2018(07):062, 2018. [arXiv:1208.2542].

[60] Ji-Ping Dai, Yang Yang, and Jun-Qing Xia. Reconstruction of the dark energy equation of state from the latest observations. *The Astrophysical Journal*, 857(1):9, 2018.

[61] SM Crawford, AL Ratsimbazafy, CM Cress, EA Olivier, S-L Blyth, and KJ Van Der Heyden. Luminous red galaxies in simulations: cosmic chronometers? *Monthly Notices of the Royal Astronomical Society*, 406(4):2569–2577, 2010.

[62] Raul Jimenez, Licia Verde, Tommaso Treu, and Daniel Stern. Constraints on the equation of state of dark energy and the hubble constant from stellar ages and the cosmic microwave background. *The Astrophysical Journal*, 593(2):622, 2003. [arXiv: astro-ph/0302560].

[63] Joan Simon, Licia Verde, and Raul Jimenez. Constraints on the redshift dependence of the dark energy potential. *Physical Review D*, 71(12):123001, 2005. [arXiv: astro-ph/0412269].

[64] Daniel Stern, Raul Jimenez, Licia Verde, Marc Kamionkowski, and S Adam Stanford. Cosmic chronometers: constraining the equation of state of dark energy. i: $h(z)$ measurements. *Journal of Cosmology and Astroparticle Physics*, 2010(02):008, 2010. [arXiv:0907.3149].

[65] Michele Moresco, Licia Verde, Lucia Pozzetti, Raul Jimenez, and Andrea Cimatti. New constraints on cosmological parameters and neutrino properties using the expansion rate of the universe to $z \sim 1.75$. *Journal of Cosmology and Astroparticle Physics*, 2012(07):053, 2012. [arXiv:1201.6658].

[66] Michele Moresco. Raising the bar: new constraints on the hubble parameter with cosmic chronometers at $z \sim 2$. *Monthly Notices of the Royal Astronomical Society: Letters*, 450(1):L16–L20, 2015. [arXiv:1503.01116].

[67] Michele Moresco, Lucia Pozzetti, Andrea Cimatti, Raul Jimenez, Claudia Maraston, Licia Verde, Daniel Thomas, Annalisa Citro, Rita Tojeiro, and David Wilkinson. A 6% measurement of the hubble parameter at $z \sim 0.45$: direct evidence of the epoch of cosmic re-acceleration. *Journal of Cosmology and Astroparticle Physics*, 2016(05):014, 2016. [arXiv:1601.01701].

[68] AL Ratsimbazafy, SI Loubser, SM Crawford, CM Cress, BA Bassett, RC Nichol, and P Väisänen. Age-dating luminous red galaxies observed with the southern african large telescope. *Monthly Notices of the Royal Astronomical Society*, 467(3):3239–3254, 2017. [arXiv:1702.00418].

[69] Julien Guy, P Astier, S Baumont, D Hardin, R Pain, N Regnault, S Basa, RG Carlberg, A Conley, S Fabbro, et al. Salt2: using distant supernovae to improve the use of type ia supernovae as distance indicators. *Astronomy & Astrophysics*, 466(1):11–21, 2007. [arXiv:astro-ph/0701828].

[70] Marisa Cristina March. *Advanced statistical methods for astrophysical probes of cosmology*. Springer Science & Business Media, 2013.

[71] Natallia V Karpenka. The supernova cosmology cookbook: Bayesian numerical recipes. *arXiv preprint arXiv:1503.03844*, 2015.

[72] Pierre Astier. The expansion of the universe observed with supernovae. *Reports on Progress in Physics*, 75(11):116901, 2012.

[73] Khaled Said, Matthew Colless, Christina Magoulas, John R Lucey, and Michael J Hudson. Joint analysis of 6dfgs and sdss peculiar velocities for the growth rate of cosmic structure and tests of gravity. *Monthly Notices of the Royal Astronomical Society*, 497(1):1275–1293, 2020. [arXiv:2007.04993].

[74] Nick Kaiser. Clustering in real space and in redshift space. *Monthly Notices of the Royal Astronomical Society*, 227(1):1–21, 1987.

[75] Luca Amendola, Martin Kunz, and Domenico Sapone. Measuring the dark side (with weak lensing). *Journal of Cosmology and Astroparticle Physics*, 2008(04):013, 2008. [arXiv:0704.2421].

[76] Bryan Sagredo, Savvas Nesseris, and Domenico Sapone. Internal robustness of growth rate data. *Physical Review D*, 98(8):083543, 2018. [arXiv:1806.10822].

[77] Joshua S Speagle. dynesty: a dynamic nested sampling package for estimating bayesian posteriors and evidences. *Monthly Notices of the Royal Astronomical Society*, 493(3):3132–3158, 2020. [arXiv:1904.02180].

[78] Daniel Foreman-Mackey, David W Hogg, Dustin Lang, and Jonathan Goodman. emcee: the mcmc hammer. *Publications of the Astronomical Society of the Pacific*, 125(925):306, 2013. [arXiv:1202.3665].

[79] Antony Lewis. Getdist: a python package for analysing monte carlo samples. 2019. [arXiv:1910.13970].

[80] Daniel Foreman-Mackey. corner. py: Scatterplot matrices in python. *The Journal of Open Source Software*, 1:24, 2016.

[81] Will Handley. fgivenx: Functional posterior plotter. *The Journal of Open Source Software*, 3(28), Aug 2018.

[82] Philip Graff, Farhan Feroz, Michael P Hobson, and Anthony Lasenby. Bambi: blind accelerated multimodal bayesian inference. *Monthly Notices of the Royal Astronomical Society*, 421(1):169–180, 2012. [arXiv:1110.2997].

[83] Philip Graff, Farhan Feroz, Michael P Hobson, and Anthony Lasenby. Skynet: an efficient and robust neural network training tool for machine learning in astron-

omy. *Monthly Notices of the Royal Astronomical Society*, 441(2):1741–1759, 2014. [arXiv:1309.0790].

[84] Will Handley. pyBAMBI. https://pybambi.readthedocs.io/en/latest/#, 2018. [Online: accessed 9-January-2020].

[85] F Feroz, MP Hobson, and M Bridges. Multinest: an efficient and robust bayesian inference tool for cosmology and particle physics. *Monthly Notices of the Royal Astronomical Society*, 398(4):1601–1614, 2009. [arXiv:0809.3437].

[86] N Aghanim, Yashar Akrami, M Ashdown, J Aumont, C Baccigalupi, M Ballardini, AJ Banday, RB Barreiro, N Bartolo, S Basak, et al. Planck 2018 results. vi. cosmological parameters. 2018. [arXiv:1807.06209].

[87] Adam G Riess, Lucas M Macri, Samantha L Hoffmann, Dan Scolnic, Stefano Casertano, Alexei V Filippenko, Brad E Tucker, Mark J Reid, David O Jones, Jeffrey M Silverman, et al. A 2.4% determination of the local value of the hubble constant. *The Astrophysical Journal*, 826(1):56, 2016. [arXiv:1604.01424].

[88] JA Vazquez, I Gomez-Vargas, and A Slosar. Updated version of a simple mcmc code for cosmological parameter estimation where only expansion history matters. https://github.com/ja-vazquez/SimpleMC, 2020.

[89] Henry W Leung and Jo Bovy. Deep learning of multi-element abundances from high-resolution spectroscopic data. *Monthly Notices of the Royal Astronomical Society*, 483(3):3255–3277, 2019. [arXiv:1808.04428].

[90] J Alberto Vázquez, David Tamayo, Anjan A Sen, and Israel Quiros. Bayesian model selection on scalar $\epsilon$-field dark energy. *Phys. Rev. D*, 103(4):043506, 2021. [arXiv:2009.01904].

# LIST OF ABBREVIATIONS

$\Lambda$**CDM** Lambda Cold Dark Matter.

**AE** Autoencoder.

**ANN** Artificial Neural Network.

**BAO** Baryon Acoustic Oscillations.

**CC** Cosmic chronometers.

**CMB** Cosmic Microwave Background.

**DE** Dark Energy.

**DO** dropout.

**FFNN** Feed-forward Neural Network.

**iid** independent and identically distributed.

**JLA** Joint Light-curve Analysis.

**MC-DO** Monte Carlo dropout.

**MSE** Mean Squared Error.

**PCA** Principal Component Analysis.

**RELU** Rectified Linear Unit.

**SNeIa** Supernovae IA.