Università degli Studi di Trieste

Dipartimento di Fisica

Corso di Laurea Magistrale in Fisica Nucleare e Subnucleare

Tesi di Laurea Magistrale

Regression Deep Neural Networks for top-quark-pair resonance finding in the dilepton channel: feasibility study and foreseen improvements over traditional analysis techniques with the ATLAS experiment at the LHC

Laureando:
Daniele Tilotta

Relatore:
Dott. Michele Pinamonti
Correlatore:
Prof.ssa Marina Cobal

ANNO ACCADEMICO 2019–2020

Abstract

The ATLAS experiment at the LHC has the potential to find evidences of new physics beyond the Standard Model (BSM). Some BSM theories predict the existence of new massive particles decaying to pairs of top quarks ($t\bar{t}$), that can be detected by ATLAS. The key distribution to consider to prove the existence of such hypothetical particles is the invariant mass of the reconstructed $t\bar{t}$ pair, $m_{t\bar{t}}$. The reconstruction of such an invariant mass from the detected particles in each event is particularly hard in the dilepton $t\bar{t}$ decay channel, where the final states include two neutrinos, which always escapes undetected. Several traditional analysis techniques already exist, but a new data analysis tool could improve the accuracy of the $m_{t\bar{t}}$ estimate. This thesis consists of a feasibility study, in which deep learning, an innovative multivariate analysis technique, is applied to simulated ATLAS events to perform a regression task. A deep neural network is successfully implemented in the feed-forward architecture and it proves to be a useful tool for reconstructing the invariant mass of top-quark pairs decaying in the dilepton channel. Comparing this innovative method with some of the most commonly used traditional analysis techniques, it was possible to show significant performance improvements in both the SM and BSM scenarios.

Sommario

L'esperimento ATLAS all'LHC ha le potenzialità per rivelare segnali di nuova fisica oltre il Modello Standard (BSM). Alcune teorie BSM predicono l'esistenza di particelle massive che decadono in coppie di quark top ($t\bar{t}$), che possono essere rivelate da ATLAS. Per dimostrare l'esistenza di queste ipotetiche particelle, una distribuzione chiave da considerare è quella della massa invariante delle coppie $t\bar{t}$, $m_{t\bar{t}}$. Nel canale dileptonico di decadimento di tale coppia, la ricostruzione della massa invariante a partire dalle particelle rivelate in ciascun evento è un calcolo particolarmente complesso, perché gli stati finali includono due neutrini, che non possono essere rivelati. Si hanno a disposizione diverse tecniche di analisi dati per affrontare questo problema, ma un nuovo approccio potrebbe migliorare l'accuratezza della stima di $m_{t\bar{t}}$. Questa tesi consiste in uno studio di fattibilità, in cui si valuta la possibilità di applicare il deep learning (una tecnica innovativa di analisi multivariata) per effettuare una regressione su dati ATLAS simulati. Una deep neural network è stata implementata con successo nell'architettura feed-forward e ha dato prova di essere uno strumento utile per ricostruire la massa invariante di coppie di quark top che decadono nel canale dileptonico. Confrontando questo metodo innovativo con alcune delle tecniche tradizionali di analisi dati più comunemente utilizzate, è stato possibile mostrare miglioramenti significativi nelle prestazioni in entrambi gli scenari SM e BSM.

Acknowledgements

First of all, I would like to express my sincere gratitude to my supervisors, Dr Michele Pinamonti and Prof Marina Cobal, for their support and guidance during all phases of this thesis.
My appreciation also goes to the Udine/ICTP ATLAS collaboration. I am grateful for you interest and for the helpful comments.
Without rethoric and without listing, thanks to everybody who encouraged me and showed interest in my work over the years.

I would like to thank John Coltrane and the city of Erice for the inspiration.

Contents

Chapter 1

Introduction

The experimental program of the Large Hadron Collider (LHC) has the potential to address many of the most fundamental questions in modern particle physics, such as the unification of the fundamental forces, the dimensionality of space, the matter/antimatter asymmetry and the particle nature of dark matter, to name a few. Some of these questions cannot be answered within the Standard Model (SM), which is currently the most successful, but incomplete, theory in particle physics, and this gave rise to many Beyond the Standard Model (BSM) theories. Within the SM, the top quark is the heaviest known fundamental particle in nature and plays an important role in many BSM theories. Some of these theories predict the existence of new massive particles decaying to pairs of top quarks, $t\bar{t}$. This kind of particles could be produced at the LHC and revealed by the LHC experiments such as ATLAS [1] and CMS [2], by reconstructing the decay products of the $t\bar{t}$ pair. Once collision events have been identified as top-quark-pair production events, a useful quantity to consider to prove the existence of such new particles is the invariant mass of the reconstructed $t\bar{t}$ pair, whose expression is given by:

$$m_{t\bar{t}} = \sqrt{(p^{\bar{t}} + p^{t})_{\mu}(p^{\bar{t}} + p^{t})^{\mu}} \tag{1.1}$$

where p^t is the four-momentum of the top quark, $p^{\bar{t}}$ is the four-momentum of the antitop quark and $\mu = 0, ..., 3$. As no $t\bar{t}$ bound states are predicted by the SM, at high values the $m_{t\bar{t}}$ distribution is expected to be exponentially decaying, whereas in new physics models a resonance bump at a given mass can appear on top of the SM prediction. The reconstruction of such discontinuities in the $m_{t\bar{t}}$ spectrum would constitute a clear evidence of new particles.
Traditional data analysis techniques in High-Energy Physics (HEP) use a sequence of boolean decisions (event selection criteria and construction of the observable distribution trough certain combinations of the quantities measured in the detector), followed by a statistical analysis of the selected data. Typically, both the individual decisions and the subsequent statistical analysis are based on the distribution of a single observed quantity motivated by physics considerations, which is not easily extendable to higher dimensions. The aim of this thesis is to study an innovative method, based on the usage of multivariate techniques and especially the so-called Deep Neural Networks (DNN),

to estimate the value of $m_{t\bar{t}}$ in each event based on all the measured quantities and to show the sensitivity improvements that it could bring to searches for new physics signals. Deep Learning (DL) is a recently developed subset of Machine Learning (ML) algorithms, which have played an important role in the analysis of HEP data for decades. ML algorithms improve automatically their performance through experience, without being explicitly programmed to perform a specific task and with few or no assumptions on the underlying physical processes. The emergence of DL in 2012 allowed to handle higher dimensional and more complex problems than previously feasible. DL refers in particular to a broad class of ML methods emphasizing hierarchical representations of the data. Several studies have demonstrated that the traditional *shallow* networks, based on physics-inspired engineered (high-level) features, are outperformed by *deep* networks, based on the higher dimensional features which receive less or no pre-processing (lower-level features).
Machine learning is nowadays widely used in experimental particle physics, for various tasks. In particular, within the ATLAS experiment, ML techniques are used for particle identification such as electron ID [3] and b-tagging [4], for which DL is used, and for signal-versus-background discrimination (e.g. [5] and [6]). These applications all belong to the category of the *classification* tasks: particle candidates or recorded collision events are accepted or rejected, or assigned to a particular bin of a histogram, based on the output of a multivariate discriminant, trained to provide an optimal discrimination between two distinct categories. *Regression* tasks differ from classification tasks since their aim is to reconstruct a continuous quantity instead of providing a binary answer. These tasks are also more and more performed through ML in ATLAS and CMS, in particular to obtain particle energy calibration, for instance for photons [7], tau-leptons [8] or b-jets [9]. Until now, the usage of regression via ML for event-level tasks such as resonance reconstruction from final-state particles is not common in the experimental collaborations, and mostly isolated studies are performed [10]. It could instead be a useful tool for various tasks, such as differential cross-section measurements or searches for new heavy particles by reconstructing their decay from the final state particles, and this thesis wants to study one of these applications.

In the SM theoretical frame, top quarks are predicted to decay to a W-boson and a b-quark nearly 100% of the times. Events with a $t\bar{t}$ pair can then be classified as *all hadronic*, *single-lepton* or *dilepton*, according to the decay of the two W-bosons: a pair of quarks or a charged lepton-neutrino pair. In this thesis, only $t\bar{t}$ pairs decaying to dilepton final states are considered. Events in this channel are hard to reconstruct exactly, due to the partial information available. These complex final states always include hadronic jets (difficult to distinguish from other products of the collision event) and neutrinos, which escape undetected. All the data analyzed consist of Monte Carlo (MC) generated samples that simulate the production of $t\bar{t}$ events and the detection of their decay products by the ATLAS experiment. Thanks to the fact that in these simulated events the generated value of $m_{t\bar{t}}$ is known, it is possible to use these samples to train the DNN and to test its performance, also comparing

them with more traditional data analysis methods.

The chapters in this thesis are organised as follows. Chapter 2 gives an introduction to the SM, with a particular attention to the top quark physics, and gives also a brief overview to some BSM theories. Chapter 3 illustrates the main features of the LHC and one of its general purpose detectors, the ATLAS detector. Chapter 4 is dedicated to the theory of DL algorithms, once the basic concepts in ML are provided. Among the many ML classes of algorithms, only the artificial neural networks are discussed, and in particular the feed-forward architecture. Chapter 5 reviews some of the most popular learning algorithms, that allow neural networks to learn from data to produce a desired result. After dealing with the theory, the concepts illustrated so far are applied to the analysis of the simulated data. The performance of the DNN are evaluated both on SM and BSM simulated event samples and compared to other analysis methods. Finally, the conclusions are given in Chapter 7.

Chapter 2

Theoretical Background

The desire to describe natural phenomena has driven scientists to formulate, over the centuries, more and more accurate mathematical models, based on few principles or assumptions. In particle physics, the most successful theory is currently the Standard Model (SM), which summarizes the theoretical foundations of modern particle physics and describes properties and interactions of such elementary particles. From its debut in the second half of the 20^{th} century, the SM has shown a strong predictive power, in excellent agreement with experimental evidences from particle colliders, such as LEP and the Tevatron. Although modern experiments at the LHC have largely reaffirmed its high compatibility with collected data, there are reasons to believe that it might not be a complete model. A quantum description of the gravitational force is not yet included and therefore the idea of unification between all fundamental interactions cannot be accomplished within the SM. Furthermore, neutrino masses and oscillations, just to mention a few, are not predicted. This gave rise to many Beyond the Standard Model (BSM) theories, that try to find and describe new areas in physics that are not covered by the SM, making predictions of the existence of new particles that arise from the introduction of new fields and symmetries. In this chapter, an overview of the SM is given in Section 2.1, with a particular attention to the top quark production and decay mechanisms (Section 2.2). Finally, a short description of some BSM theories, emphasizing the role of the top quark, is given in Section 2.3.

2.1 The Standard Model

The standard model of particle physics describes elementary particles and their interactions. The mathematical ground upon which the SM is rigorously built is provided by Quantum Field Theory (QFT), a framework that reconciles quantum mechanics and special relativity, in which particles can be described by quantum fields. The interaction between particles is governed by Quantum Electro-Dynamics (QED), which describes the electroweak interaction processes (such as beta decay, leptons interaction, heavy leptons decay, etc.) and Quantum Chromo-Dynamics (QCD), that describes the behaviour of strongly interacting particles. The model itself also includes a mass generating

mechanism, referred to as the Higgs mechanism.

2.1.1 Elementary Particles

Elementary particles are fundamental objects that have no internal structure. Like a periodic table, the SM classifies such particles according to their properties (Fig. 2.1). A first classification is based on the spin, which groups fermions and bosons. Fermions are spin 1/2 particles that obey to Fermi-Dirac statistics and that constitute, as matter, $\sim 5\%$ of the universe [11]. Depending on how they interact through the strong, weak and electromagnetic force, the fermions are further divided in six quarks and six leptons, which occur in pairs, differing by one unit of electric charge e, and are replicated in three generations with a strong hierarchy in mass.

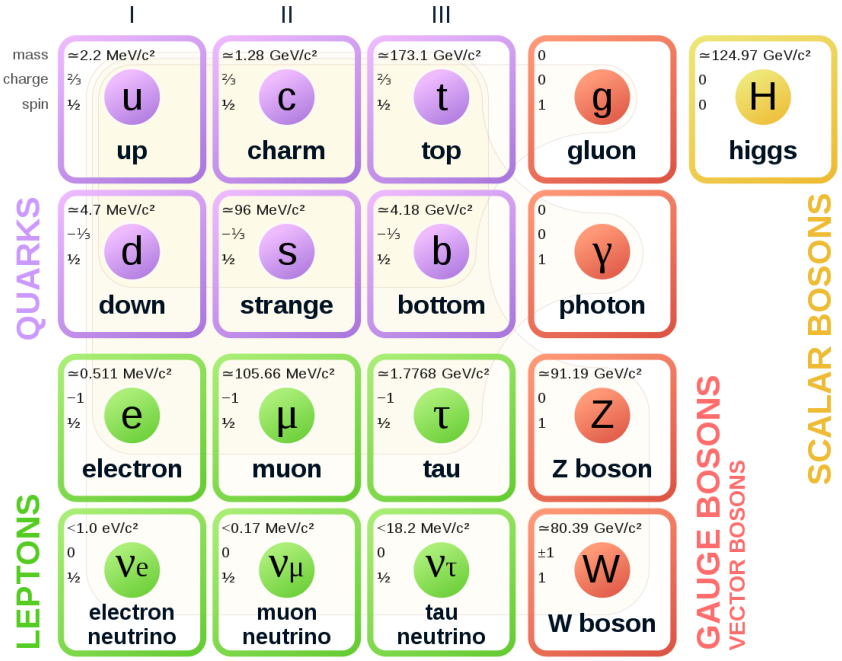

Figure 2.1: The SM includes 12 fundamental fermions and 5 fundamental bosons. In this picture, name and properties (mass, charge, spin) of all known particles are listed. Quarks and leptons are arranged in three generations from left to right. The first row of quarks shows up-type quarks, the second down-type quarks. Brown loops indicate which bosons (red) couple to which fermions (purple and green).

Each quark generation $(u, d), (c, s), (t, b)$ contains a positive quark with charge $2/3$ e and a negative quark with charge $-1/3$ e, while a lepton generation consists of a particle with a negative unit charge (e, μ, τ) and a corresponding zero charge neutrino $(\nu_e, \nu_\mu, \nu_\tau)$. Besides the electric charge and the spin, the fermions also have other quantum numbers. The coupling to the weak interaction is associated with the weak isospin, related to the chirality which characterizes the handedness of a fermion, corresponding to a left-handed and right-handed projection. The quarks have an additional color charge, labeled

Interaction	Theory	Mediator	Strength	Behavior	Range
Weak	EW	W and Z	10^{25}	$\frac{1}{r}e^{-m_{W,Z}r}$	10^{-18} m
Strong	QCD	Gluon	10^{38}	$\sim r$	10^{-15} m
EM	QED	Photon	10^{36}	$\frac{1}{r}$	∞
Gravity	GR	Graviton	1	$\frac{1}{r}$	∞

Table 2.1: Summary of the main characteristics of the four fundamental forces.

into *red*, *green* and *blue*, which represents the coupling with the strong interaction. Via the strong interaction, quarks can form bound color-singlet states called hadrons, consisting of either a quark and an antiquark (*mesons*) or three quarks (*baryons*). The fact that only color-neutral states and no free quarks are observed in nature is referred to as the *confinement* of quarks in hadrons[1]. According to the CPT theorem, each fermion has its anti-particle with opposite quantum numbers, but with the same mass.
On the other hand, bosons are integer-spin particles that obey to the Bose-Einstein statistics. Gauge bosons are spin 1 force carriers, that can be thought as particles that are exchanged to mediate a particular type of force among the fermions. These include the massless *photon* (γ) for electromagnetism and *gluon* (g) for the strong force, as well as the massive Z^0 and $W^{\pm}$ bosons for the weak force. Fermions cannot interact with all bosons. Neutrinos only interact via the weak interaction, charged leptons undergo both weak and electromagnetic interaction, while quarks participate in all three fundamental interactions. Finally, the Higgs boson, the most recently discovered SM particle, is the spin 0 particle required for spontaneous electroweak symmetry breaking. The Higgs boson arises from the Higgs mechanism that gives mass to the fundamental particles, as described in Section 2.1.2.

2.1.2 Fundamental Interactions

The SM also describes the interactions between the particles just mentioned. According to the present understanding, there are four fundamental interactions or forces known to exist, that do not appear to be reducible to more basic interactions: gravitation, electromagnetism, the weak interaction, and the strong interaction. Their magnitude and behaviour vary greatly, as schematized in Table 2.1. The gravitational and electromagnetic interactions produce significant long-range forces, whose effects can be seen directly in everyday life, while the strong and weak interactions produce forces at subatomic distances and govern nuclear interactions. Each of the known fundamental interactions can be described mathematically as a field. The gravitational force is attributed to the curvature of spacetime, described by Einstein's general theory of relativity. In theories of quantum gravity, the graviton is the hypothetical elementary particle that mediates the force of gravity. No theory that attempts

[1]With the exception of the top quark (see Section 2.2)

to include gravity in the SM has successfully been verified by the experiments, but this is mostly inconsequential in practice, due to the weakness of gravity compared to the other forces, which are instead discrete quantum fields, and their interactions are mediated by elementary bosons described by the SM. The Lagrangian formalism is used to codify kinetic and potential terms that regulate the possible interactions among fields and their relative strength. In the language of group theory, the SM obeys an $SU(3) \times SU(2) \times U(1)$ symmetry. The $SU(3)$ group arises from the theory of Quantum Chromo-Dynamics (Sec. 2.1.2), the SU(2) and U(1) group from Electroweak Theory (Section 2.1.2).

Quantum Chromo-Dynamics

The Quantum Chromo-Dynamics (QCD) is an $SU(3)$ group based theory which describes strong interactions. Considering that quarks are spin 1/2 particles that obey Fermi-Dirac statistics, in order to have an anti-symmetric total wave function, a new quantum number must be introduced. Each quark can occur in one of the three color charge states *red* (r), *green* (g) and *blue* (b), while anti-quarks can have three different anti-colors. The gauge bosons, which are eight gluons, carry $(r + b + g) \otimes (\bar{r} + \bar{b} + \bar{g}) - (r\bar{r} + b\bar{b} + g\bar{g})$ color charges, hence they can have self-interactions. The only bound states invariant under the $SU(3)$ transformations and that can be observed experimentally are the colorless bound states, the *hadrons*, which are composed by point-like constituents, termed *partons*, which describe the same objects now more commonly referred to as quarks and gluons. As already mentioned, hadrons are then re-grouped in two configuration: baryons ($\bar{q}\bar{q}\bar{q}$ or qqq) and mesons ($q\bar{q}$). The QCD Lagrangian, which describes the strong interactions, is:

$$\mathcal{L} = \sum_q i\bar{q}\gamma^\mu D_\mu q - \frac{1}{4} G^\alpha_{\mu\nu} G^{\mu\nu}_\alpha \tag{2.1}$$

where q is the quark field, $\bar{q}$ is the anti-quark field and $G^{\mu\nu}_\alpha$ is the gluon field. The covariant derivative for the $SU(3)$ symmetry group is given by:

$$D_\mu = \partial_\mu + i g_s T_a G^a_\mu \tag{2.2}$$

where T_a, the generators, are three-dimensional matrices with $a = 1, ..., 8$ which can be expressed in terms of the Gell-Mann matrices. The g_s is a coupling constant, related to the coupling strength α_s by:

$$\alpha_s = \frac{g_s^2}{4\pi} \tag{2.3}$$

The quantity α_s is not indeed a real constant, but it depends on the energy scale. At high energies (small distances) it decreases, which means that quarks can be considered as free particles and this property is referred to as *asymptotic freedom*. On the other hand, partons cannot appear as free particles at large distances. When trying to separate quarks, a pair of quarks with opposite color charge will be created from the vacuum, and new bound

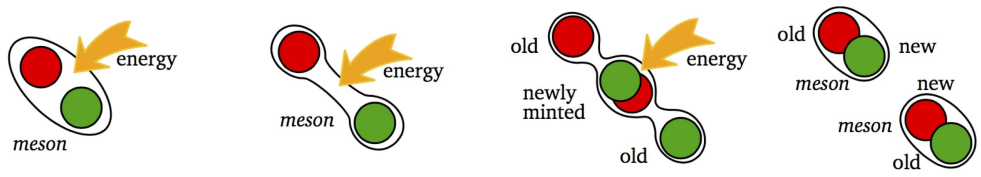

Figure 2.2: Schematic representation of the color confinement. In the attempt to separate two quarks, a new bound state is created.

states will appear. This phenomenon, called *color confinement*, is illustrated in Fig. 2.2. When studying high-energy collisions, one often has to consider processes where quarks and gluons are produced in the final state. However, these particles are not directly observable. First of all, they tend to undergo successive branchings at small angles, producing a series of collimated quarks and gluons. The parton shower (PS) is a phenomenon that refers to partons above $\simeq$ 1 GeV which undergo QCD radiation of gluons and photons. Starting from a parton with high virtuality ($Q^2 = -q^2$, where q is the four-momentum of the parton), of the order of the hard scale of the process, the parton shower will produce branchings into further partons of decreasing virtuality, until one reaches a non-perturbative hadronization scale. Through radiation, quarks and gluons degrade their energies and their interactions become stronger and stronger, until, due to confinement, they cluster together to form colorless baryons and mesons, which might be excited and also decay into many lower-energy states. The transformation of partons into hadrons, commonly referred to as the *hadronization* process, does not significantly alter the energy–momentum flow of the original quarks and gluons. Overall, the high-energy partons produced by the primary collision appear in the final state as bunch of hadrons, called *jets*, that are highly collimated in the directions of the original partons (see Fig. 2.3).
This behaviour is observed directly in experiments where the hadronic final state appears to be collimated around a few directions in the detector, which represent the footprints of unobservable quarks and gluons.

Electroweak Theory

The theory of Electroweak (EW) interactions, proposed in the sixties by Glashow, Salam and Weinberg, unify the theory of Quantum Electrodynamics (QED) and the Weak Theory (WT), describing electromagnetic and weak forces as two low-energy manifestations of a more fundamental interaction. This theory is invariant under the transformations of the gauge group $SU(2)_L \times U(1)_Y$. The first part of the symmetry group $SU(2)_L$ introduces the so called weak isospin, T, which can be written in the form of $T_i = \frac{\tau_i}{2}$ $(i = 1, 2, 3)$, where τ_i are the Pauli matrices, corresponding to three massless gauge fields W_{μ^i}. Since the weak interaction only couples to left-handed particles, the fermion

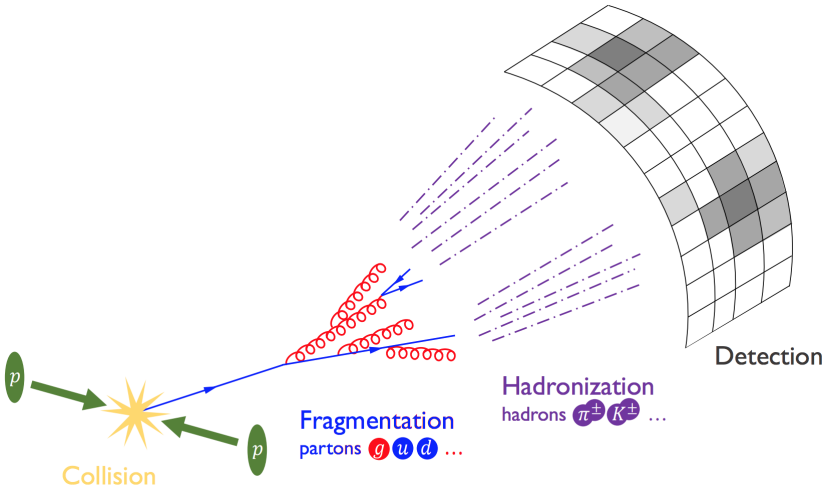

Figure 2.3: Schematic view of the jet formation process. Quarks originating from the beams collision are fragmented into a fractal of quarks and gluons, which hadronize into the observed particles.

fields ψ are split up into left-handed (ψ_L) and right-handed (ψ_R) fields:

$$\psi_L = \frac{1}{2}(1-\gamma^5)\psi$$
$$\psi_R = \frac{1}{2}(1+\gamma^5)\psi \tag{2.4}$$

where $\frac{1}{2}(1\pm\gamma^5)$ are the chirality operators, with $\gamma^5 = i\gamma^0\gamma^1\gamma^2\gamma^3$, and γ^i are the Dirac matrices. The left-handed fermions form weak isospin doublets ($T = \frac{1}{2}$) and the right-handed fermions are isospin singlets ($T = 0$):

$$\begin{pmatrix} u \\ d \end{pmatrix}_L \begin{pmatrix} \nu_e \\ e \end{pmatrix}_L \quad \begin{pmatrix} c \\ s \end{pmatrix}_L \begin{pmatrix} \nu_\mu \\ \mu \end{pmatrix}_L \quad \begin{pmatrix} t \\ b \end{pmatrix}_L \begin{pmatrix} \nu_\tau \\ \tau \end{pmatrix}_L \tag{2.5}$$

$$u_R, d_R, e_R \quad c_R, s_R, \mu_R \quad t_R, b_R, \tau_R \tag{2.6}$$

Left-handed fermions and right-handed anti-fermions interact via the exchange of the gauge bosons $W^\pm$ and Z^0. In the doublets, neutrinos and up-type quarks (u, c, t) have the weak isospin $T_3 = +1/2$, while the charged leptons and down-type quarks (d, s, b) carry the weak isospin $T_3 = -1/2$. The right-handed fermions and left-handed anti-fermions interact via the exchange of a Z^0 boson and γ only.
On the other hand, the second part of the symmetry group $U(1)_Y$, introduces the hypercharge Y, and corresponds to one massless gauge field B_μ. The hypercharge is associated with the electric charge and the third component of the weak isospin T_3 via the Gell-Mann Nishijima formula:

$$Y = 2(Q - T_3) \tag{2.7}$$

The full Lagrangian of the electroweak interactions can be written as:

$$\mathcal{L}_{EW} = \psi_L^\dagger i\gamma^\mu D_\mu \psi_L + \psi_R^\dagger i\gamma^\mu D_\mu \psi_R - \frac{1}{4}W^i_{\mu\nu}W_i^{\mu\nu} - \frac{1}{4}B_{\mu\nu}B^{\mu\nu} \tag{2.8}$$

where the first two terms describe particle interactions, while the last two terms are related to the gauge field interactions. The covariant derivative in the electroweak interactions for ψ_L fermions is given by

$$D^\mu = \partial^\mu + ig\frac{\tau_i}{2}W_\mu^i + i\frac{g'}{2}YB^\mu \tag{2.9}$$

while for ψ_R fermions, the covariant derivative is

$$D^\mu = \partial^\mu + i\frac{g'}{2}YB^\mu \tag{2.10}$$

where g and g' are the coupling constants for $SU(2)$ and $U(1)$ respectively. The physics fields can be derived using the covariant derivatives and from the linear combinations of the gauge fields W_μ and B_μ

$$A_\mu = W_\mu^3 sin(\theta_W) + B_\mu cos(\theta_W) \tag{2.11}$$

$$Z_\mu = W_\mu^3 cos(\theta_W) - B_\mu sin(\theta_W) \tag{2.12}$$

$$W_\mu^\pm = \frac{1}{\sqrt{2}}(W_\mu^1 \mp W_\mu^2) \tag{2.13}$$

where A_μ, Z_μ and $W_\mu^\pm$ correspond to the photon, Z and W boson fields, respectively. In the previous equations, θ_W is the Weinberg angle, which describes the mixing between $SU(2)$ and $U(1)$, and is defined in terms of the weak and electromagnetic coupling constants as:

$$\sin(\theta_W) = \frac{g'}{\sqrt{(g^2 + g'^2)}} \tag{2.14}$$

The electroweak theory also describes the weak interactions between quarks from different generations through the exchange of a charged W boson. In the quark sector, the mixing between weak eigenstates of the down-type quarks d', s' and b', and the corresponding mass eigenstates d, s and b, is described by the Cabibbo-Kobayashi-Maskawa (CKM) matrix [12] [13]:

$$\begin{pmatrix} d' \\ s' \\ b' \end{pmatrix} = \begin{pmatrix} V_{ud} & V_{us} & V_{ub} \\ V_{cd} & V_{cs} & V_{cb} \\ V_{td} & V_{ts} & V_{tb} \end{pmatrix} \begin{pmatrix} d \\ s \\ b \end{pmatrix} \tag{2.15}$$

where $|V_{ij}|$ is the probability of transition between q_i up-type and q_j down-type quarks via the exchange of a $W^\pm$ boson. By convention, the mixing takes place between down-type quarks only, since the CKM matrix is not diagonal, while the up-type mass matrix is diagonal. This unitary matrix has diagonal entries close to unity, which means that the probability of transition between quarks of the same generation is dominating. Off-diagonal entries are around 0.2 between the first and second generation, around 0.04 between the second and third generation and even smaller for the transition of the first to the third generation [13]. In particular, the matrix element V_{tb} is indirectly constrained to be very close to 1 by making use of the unitarity of the CKM

matrix and assuming three quark generations. Hence, the top quark in the SM is forced to couple almost exclusively to bottom quarks. This affects both the top quark production, suppressing the electroweak single top production mechanisms with respect to the pair production one (see Sec. 2.2.1 and 2.2.2), and its decay, allowing to rely on the presence of b-quark jets in the final state to isolate and reconstruct top events (see Sec. 2.2.3).
In the lepton sector, if the neutrinos are assumed to be massless, such a mixing does not take place. However, from experimental evidences, neutrinos also have mass, which has led, among other things, to the introduction of an analogue leptonic mixing matrix, the Pontecorvo-Maki-Nakagawa-Sakata (PMNS) matrix [14]. The gauge bosons (A_μ, Z_μ and $W^\pm_\mu$) and the fermions in this model are massless particles. However, this is not in agreement with experiments, that confirmed the existence of massive fermions and electroweak mediators with masses of $m_W = 80.4$ GeV and $m_{Z^0} = 91.2$ GeV [15]. Including the gauge boson mass terms in the electroweak theory violate the gauge invariance. A minimal way to incorporate these observed masses is to implement electroweak Spontaneous Symmetry Breaking (SSB) at energies around the mass scale of the W and Z-bosons, often referred to as the *Higgs mechanism* [16], by introducing a new scalar field to the electroweak Lagrangian.

The Higgs Mechanism

The Higgs mechanism breaks down the symmetry group $SU(2)_L \times U(1)_Y$ to $U(1)_{EM}$. An additional scalar field ϕ is introduced, named *Higgs field*, which is composed by a doublet of two complex scalar fields in $SU(2) \times U(1)$ representation:

$$\phi = \frac{1}{\sqrt{2}} \begin{pmatrix} \phi_1 + i\phi_2 \\ \phi_3 + i\phi_4 \end{pmatrix} \tag{2.16}$$

The Lagrangian for this field is

$$\mathcal{L}_H = (D_\mu \phi)^\dagger (D^\mu \phi) - V(\phi) \tag{2.17}$$

which consists of a kinetic term and a *Higgs potential* $V(\phi)$

$$V(\phi) = \mu^2 \phi^\dagger \phi + \lambda (\phi^\dagger \phi)^2 = \mu^2 \phi^2 + \lambda \phi^4 \tag{2.18}$$

The first term in Eq. 2.18 can be associated with the mass of the field, while the second term stands for the self-interaction of the field. The parameter λ of the potential needs to be positive since the case of $\lambda < 0$ is unphysical. The parameter μ can be instead chosen freely. For $\mu^2 > 0$, the potential assumes a unique minimum at $\phi_0 = 0$, leading to a symmetric ground state under $SU(2)$. On the other hand, for $\mu^2 < 0$, the shape of the potential is modified as illustrated in Fig. 2.4 and V assumes a non-trivial minimum in $\phi_0 = \pm\sqrt{-\frac{\mu^2}{\lambda}} = \pm v$. When the neutral component obtains a non-zero vacuum expectation value, the $SU(2)_L \times U(1)_Y$ symmetry is broken to $U(1)_{QED}$, giving mass to the three electroweak gauge bosons $W^\pm$ and Z_0 while keeping the photon massless, and leaving the electromagnetic symmetry $U(1)_{QED}$ unbroken. The

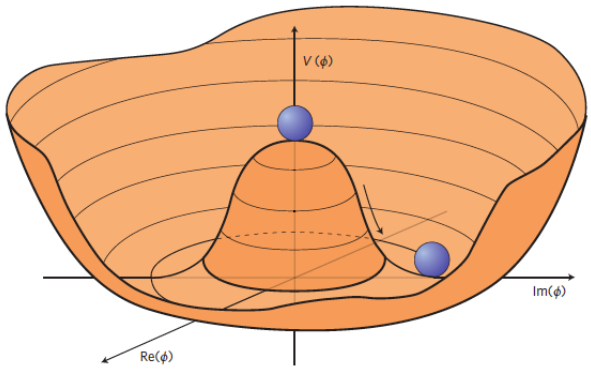

Figure 2.4: The shape of the Higgs potential $V(\phi) = \mu^2\phi^2 + \lambda\phi^4$, with $\mu^2 < 0$. The vacuum state is randomly chosen from infinite number of choices when falling into the vacuum state which leads to spontaneous symmetry breaking.

gauge bosons masses are then given in terms of the expectation value for the vacuum state, defined as the absolute value of the field at the minimum of the potential:

$$M_W = g\frac{v}{2}, \quad M_Z = \frac{v}{2}\sqrt{g^2 + g'^2}, \quad M_\gamma = 0 \tag{2.19}$$

From the remaining degree of freedom of the scalar doublet, an additional scalar particle with spin 0 and mass equal to $\sqrt{2\mu}$, the *Higgs boson*, is obtained. In 2012, the ATLAS and CMS experiments at the LHC announced for the first time the discovery of the Higgs boson, which carries a mass $m_H = 125.09 \pm 0.21(stat) \pm 0.11(sys)$ GeV and has no electric and color charge. [17]. The Higgs mechanism for the spontaneous symmetry breaking in the EW theory can also be used to generate the masses for the fermions, by adding another term to the SM Lagrangian, describing the coupling between fermions and the Higgs field.

$$\mathcal{L}_{Yukawa} = y_f \phi \bar{\psi}^f_L \psi^f_R + h.c \tag{2.20}$$

The mass for fermions is given as:

$$m_f = \frac{v}{\sqrt{2}} y_f \tag{2.21}$$

where y_f is another coupling constant referred to as Yukawa coupling, proportional to fermion mass. From this relation turns out that massive particles are those which have the most significant coupling to the higgs field. The top quark, which is the most massive elementary particle, with a mass 173.0 ± 0.4 GeV, has the largest coupling to the Higgs field with $y_{top} \simeq 1$ [17].

2.2 Top Quark Physics

The top quark was discovered in 1995 at the Tevatron collider by the CDF [18] and DØ [19] experiments. Together with the bottom quark, it belongs to the third generation of quarks. This particle has a very short lifetime $\tau_{top} \approx 5 \times 10^{-25}$ s, even shorter than the hadronisation scale. As a consequence, it decays before it can hadronize, providing a unique opportunity to study a "bare" quark. The top quark plays an essential role in the SM and in BSM theories, because, as a consequence of its high mass, it has the largest coupling constant to the Higgs field, and it is also predicted to have large coupling to many new particles. Therefore, studying and measuring the top quark properties with high precision is crucial for many BSM analyses, which predict the production of new massive particles in association with the top quark. In the SM framework, top quarks can be produced in pairs ($t\bar{t}$) via the strong interaction or singly via electroweak interaction. The two mechanisms are discussed in the following subsections, with a particular attention to the top-quark pair production process, since it is the dominant process and the simulated data analyzed in this thesis are of this type.

2.2.1 Single Top Quark Production

Top quarks can be produced singly via electroweak interactions involving the Wtb vertex. There are three production modes which are distinguished by the virtuality Q^2 of the W-boson ($Q^2 = -q^2$, where q is the four-momentum of the W):

- **t-channel**: a virtual W-boson strikes a sea b-quark inside the proton. The W-boson is space-like ($q^2 < 0$). Feynman diagrams representing this process are shown in Fig. 2.5 (a).
- **s-channel**: this production mode is of Drell-Yan type and is also called $t\bar{b}$ production. A time-like W-boson with $q^2 \geq (m_{top} + m_b)^2$ is produced by the fusion of two quarks belonging to an $SU(2)$-isospin doublet. See Fig. 2.5 (b) for the Feynman diagram.
- **W-associated production**: the top quark is produced in association with a real (or close to real) W-boson ($q^2 = M_W^2$). The initial b-quark is a sea-quark inside the proton. Fig. 2.5 (c) shows the Feynman diagram. The cross-section is negligible at the Tevatron, but of considerable size at LHC energies, where associated Wt production even exceeds the s-channel.

At LO, the cross-section for each of the production modes is proportional to the square of the CMK matrix element V_{tb}. The measurement of these production cross-section is the only direct way to measure V_{tb} without assuming the unitarity of the CKM matrix. Recently, CDF and DØ first and then ATLAS and CMS, observed the single top production, allowing a direct measurement of V_{tb} which is in good agreement with the indirect determination, suffering however from larger uncertainty.

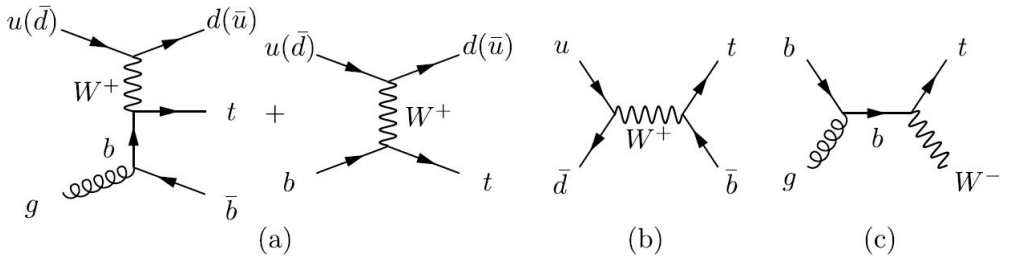

Figure 2.5: Representative Feynman diagrams for the three single top quark production modes: (a) t-channel, (c) s-channel, and (d) W-associated production process.

2.2.2 Top Quark Pair Production

Between the two possible ones, the production of a $t\bar{t}$ pair is the dominant process at both Tevatron and LHC, involving p-$\bar{p}$ and p-p collisions respectively. At the leading order (LO) in perturbation theory, top pairs can be produced by two sub-processes, either gluon-gluon (gg) fusion or quark-antiquark ($q\bar{q}$) annihilation, whose Feynman diagrams are shown in Fig. 2.6. In next-to-leading order (NLO) calculations, $t\bar{t}$ pairs are produced from quark-gluon (qg) scattering and from gluon bremsstrahlung or virtual corrections to the LO. The

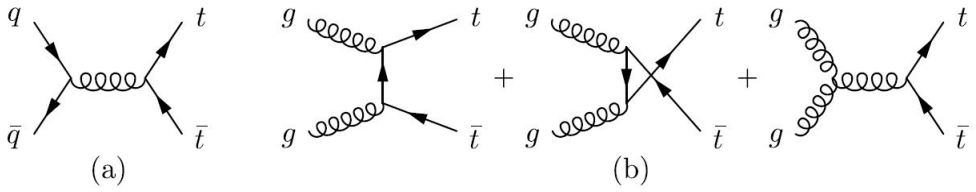

Figure 2.6: Feynman diagrams for $t\bar{t}$ production at the LO: (a) quark-antiquark annihilation ($q\bar{q} \to t\bar{t}$) and (b) gluon-gluon fusion ($gg \to t\bar{t}$).

contribution of the various processes to $t\bar{t}$ production depends on the centre-of-mass energy of the collider, as well as on the kind of colliding particles. Any hadron, including protons and antiprotons, is composed by point-like partons. The proton can be considered to accommodate three *valence* quarks (uud) which dictate its quantum numbers and typically carry much of the momentum of the proton. There are also virtual or *sea* quarks and gluons, which carry less momentum individually. When two protons (or a proton and an antiproton) collide, a hard interaction occurs between one of the partons of the first proton and one of the partons of the second proton. Soft interactions, involving the remainder of the hadron constituents, produce many low energy particles which are largely uncorrelated with the hard collision. In the centre-of-mass frame, in which the protons (or the proton and the antiproton) are rapidly moving, the hard interactions between partons are fast relative to the time for them to interact. As a result, the hadronic collision can be factorized [20] into a parton collision weighted by Parton Distribution Functions (PDFs), $F_i(x_i)$, which express the probability for the parton i to carry the momentum fraction, x_i, of its parent hadron. These PDFs are properties of

specific hadrons and are independent from the specific hard scatter interaction at parton level. A specific process production cross-section, like for example the top pair production, is then calculated as:

$$\sigma(pp \to t\bar{t} + X) = \sum_{i,j} \int dx_i dx_j \times F_i(x_i, \mu) F_i(x_j, \mu) \hat{\sigma}_{ij}(x_i, x_j, m^2_{top}, \mu^2) \quad (2.22)$$

where the sum runs over gluons and light quarks in the colliding protons, and $\hat{\sigma}_{ij}$ is the perturbative cross-section for collisions of partons i and j. The factorization scale defines the splitting of perturbative and non-perturbative elements. Usually, this scale is treated in common with the appropriate scale for the renormalization of the perturbative cross-section, since both parameters are arbitrary. This common scale, μ, is usually taken to be equal to the top quark mass (m_{top}). An exact calculation would not depend on these scales. Finite order calculations are instead sensitive to them, at a level which has to be assessed as an uncertainty in the theoretical calculation. The $\sigma_{t\bar{t}}$ has been measured by the Tevatron experiments CDF and DØ and by the LHC experiments ATLAS and CMS. The agreement between the predicted and measured values at different centre-of-mass energies for p-p and p-$\bar{p}$ collisions is shown is Fig. 2.7 [21].

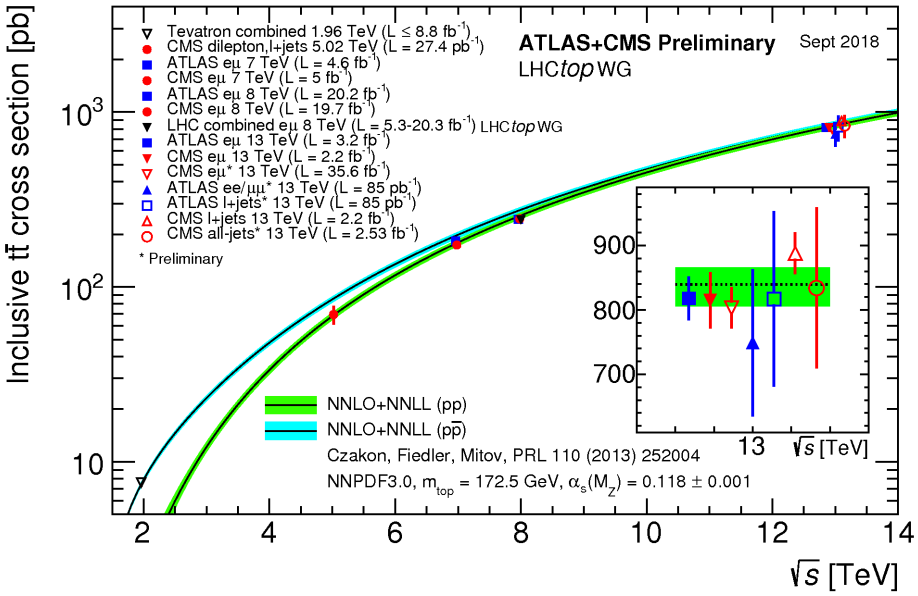

Figure 2.7: Summary of measurements for $t\bar{t}$ production cross-section as a function of $\sqrt{s}$ at the LHC and Tevatron compared with the theoretical calculation at NNLO + NNLL.

To produce the top anti-top pair, the centre-of-mass energy of the colliding partons ($\sqrt{\hat{s}}$) needs to be larger than twice the top quark mass:

$$\sqrt{\hat{s}} = \sqrt{x_i x_j s} \geq 2m_{top} \quad (2.23)$$

where x_i (x_j) is the momentum fraction of the partons participating in the collision ($p_i = x_i p_{proton}$). In the case of two partons carrying the same fraction

$x_i = x_j = x$, one obtains:

$$x \geq \frac{2m_{top}}{\sqrt{s}} \tag{2.24}$$

At the Tevatron, where $\sqrt{s} \approx 2$ TeV, the parton momentum fraction needed is $x \approx 0.18$. In this region, at energies close to the kinematic threshold, the dominant mechanism for $t\bar{t}$ production is the $q\bar{q}$ annihilation in about 85% of the cases. This happens because, in the case of $p\bar{p}$ collisions, the anti-quark is a valence quark and this kind of sub-process is more likely to occur. On the other hand, at the LHC, the anti-quark has to be a sea quark. Here, the centre-of-mass energy $\sqrt{s}$ is equal to 13 TeV, which gives $x \approx 0.03$. At this x, one finds mostly gluons, and this is why the main $t\bar{t}$ production is gg fusion, in 95% of the cases. On top of that, at Tevatron, partons carry a high fraction of the proton (anti-proton) momentum, while at the LHC small fractions x_i are enough to produce top quark pairs. For a top-quark mass of $m_t = 173.2$ GeV and with a centre-of-mass energy of 13 TeV, the measured cross section is $\sigma_{t\bar{t}} = 816.0^{+39.5}_{-44.7}$ pb [22], in which the uncertainty comes from variations of the renormalization and factorization scales as well as uncertainty associated with the PDFs.

2.2.3 Top Quark Decay

Because of its enormous mass, the top quark has an extremely short lifetime. As a result, top quarks do not have time to form hadrons before they decay, as other quarks do. Within the SM, the only known way the top quark can decay is through the weak interaction. Being heavier than the W-boson, the top quark decays, with probability close to one, to an on-shell W-boson and a b-quark.

$$t \to Wb \tag{2.25}$$

The W-boson is also an unstable particle and can decay in two channels, the leptonic and the hadronic one. A W-boson decays in about 1/3 of the cases in the leptonic channel, producing a charged lepton (e, μ, or τ) and the corresponding neutrino, with all the three lepton flavours being produced with the same probability. In the remaining 2/3 of the cases, the W-boson decays into a quark-antiquark pair, and the abundance of a given pair is instead determined by the magnitude of the relevant CKM matrix element. Specifically, the CKM mechanism suppresses the production of b-quarks as $|V_{cb}|^2 = 1.7 \times 10^{-3}$. Thus, the hadronic W-boson decay can be considered as a clean source of light quarks. The overall $t\bar{t}$ decay channels, together with the corresponding branching ratios, is shown in Fig. 2.8. From an experimental point of view, one can classify a $t\bar{t}$ pair decay into three categories, according to the number of W-bosons which decay leptonically:

- **Fully hadronic channel**: both Ws decay hadronically, giving six jets in the event: two b-jets from the top decay and four light jets from the W-boson decay. This channel is the most probable, with BR $\approx$ 46%, and its reconstruction is very challenging. Having no high p_T lepton to

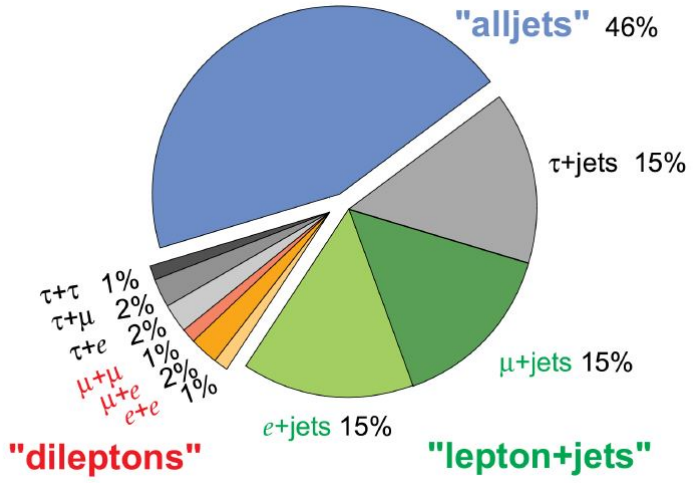

Figure 2.8: Branching ratios for the different $t\bar{t}$ decay channels.

trigger on, the signal is not easily distinguishable from the QCD multijet background, expected to be orders of magnitude bigger. Another challenging point of this signature is the presence of a high combinatorial background when reconstructing the top quark mass.

- **Single-lepton channel**: the branching ratio for this channel, where one of the W-bosons decays leptonically while the other one decays hadronically, is BR $\approx$ 45%. The $t\bar{t}$ final states consist of one lepton-neutrino pair, two light quarks and two b-quarks. The presence of a single high p_T lepton and of four or more jets allows to suppress the QCD and the W+jet backgrounds respectively. The p_T of the neutrino can be reconstructed, as it is the only source of missing energy for signal events.

- **Dilepton channel**: as both Ws decay leptonically, the final states in this channel consist of two b-quarks and two lepton-neutrino pairs. The expected BR for this channel is $\approx$ 9% and the background compared to other channels is the lowest, allowing to obtain a clean sample of top-quark events. Despite that, it is quite challenging to reconstruct the full $t\bar{t}$ event, suffering both from relatively poor statistics and from to the presence of the two neutrinos, which escape from the detector unidentified.

In the analysis presented in this thesis, only the dilepton channel is considered, without taking into account final states that involve hadronic τ lepton decays, since such particles are difficult to identify.

2.3 Beyond the Standard Model

The SM has proven to successfully describe most experimental data. With the Higgs boson discovery, all fundamental particles predicted by this theory were found. Furthermore, experimental data suggest that there is no 4^{th} generation of fermions to be expected in a mass range close to the other fermions. Despite all the successes of the SM, it is not considered as a complete theory of particle physics and there are evidences that there is physics Beyond the Standard Model (BSM) that has yet to be discovered. In the following, some of the limitations of the SM are discussed.

- **Dark matter**: many different observations indicate that there should be a different kind of stable matter that only sparsely interacts non gravitationally with SM particles or with itself. In contrast to neutrinos, those particles should have high masses to explain, among other issues, the structure formation of galaxies.

- **Matter/anti-matter asymmetry**: in the early universe, according to current cosmological models, particles, anti-particles and radiation were in constant interaction with particle creation and annihilation. As the universe expanded, the energy density dropped and the creation of particles in thermal equilibrium stopped. Since in the SM matter and anti-matter are always produced in pairs, one would expect that, when particle creation is no longer possible, particles would annihilate with the anti-particles, converting their energy to electromagnetic radiation, but this was not observed. A small amount of the matter/anti-matter imbalance can be explained with a small amount of CP violation in the SM, but for the current models of the early universe, this would not be enough to explain the asymmetry we observe.

- **The hierarchy problem**: the particle masses in the SM depend on the energy scale and undergo corrections from loop interactions with other particles. This leads, especially for the higgs boson, to high mass corrections, that are quadratic in the energy scale. If the theory is built to hold up to the highest energy scales, a huge amount of fine tuning is needed to keep the higgs mass as small as 125 GeV at the low energy regime we observe at modern particle colliders. This fine tuning is considered unnatural and other theories, like a supersymmetric extension of the SM, deliver more natural ways to explain the relatively low higgs-boson mass.

- **Neutrino masses**: in the SM, the neutrinos are assumed to be massless, but the observation of neutrino oscillations has proven that these particles actually have mass.

- **Unification**: the electromagnetic and the weak force can be described in a unified theory. In fact if the energy scale is high enough, interactions with photons and Z bosons become indistinguishable. A theory that could combine all three forces would give a more natural description of

nature. Such a unification of forces is possible in other physics models outside the scope of the SM. For example, in theories with a $SU(5)$ symmetry, the couplings of the different forces unify and that delivers additional theoretical motivation for a model beyond the SM.

2.3.1 The Top Quark in BSM Physics

Checking the internal consistency of the SM is one of the possibilities to see where new physics might arise. The dependencies of the different SM parameters on each other can be used to constrain other parameters. By comparing the fit results for one parameter with its measured value, it can be tested if the SM describes particle physics in a self consistent way. To reasonably do this, precision measurements of the different parameters are needed. One of the most influential parameters is just the mass of the heaviest SM particle, the top quark. Experimental searches for new phenomena predicted by BSM theories can also help to establish new theoretical frameworks in particle physics. Many BSM models predict the existence of new neutral or charged gauge bosons which can be produced at the LHC. These gauge bosons are massive, spin-1 particles and predicted to have the same coupling strength to the SM fermions as the SM gauge boson. Such new particles, if not too heavy, could be seen by the LHC experiments. This is in particular the case of Z' and W' bosons, which are predicted within the Sequential Standard Model (SSM) [23] and TC2 model [24], and have similar properties as the ordinary SM W and Z gauge bosons, but larger masses. The Z' boson is expected to have a mass at the TeV scale can decay to a top-antitop quark pair, as shown in Fig. 2.9.

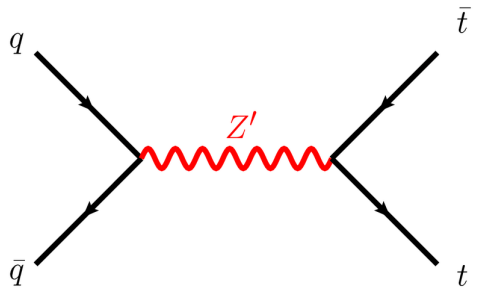

Figure 2.9: Feynman diagram for the production of Z' bosons decaying to top antitop quark pair.

The expected $m_{t\bar{t}}$ distribution is exponentially decaying under the SM prediction, while if other mechanisms of production exist, a resonance bump at given mass can appear on top of the SM distribution.

Chapter 3

Physics at the LHC

Investigating high energy physics requires the construction of machines able to explore such energy scales and to detect an adequate amount of events. In a collider, particle beams are accelerated to ultra-relativistic speeds and brought to collision in order to create heavier particles. These particles typically decay shortly after being created and leave traces in the detectors, which are meant to collect such signals. In this chapter a brief introduction to the LHC collider and its physics program is given, together with a description of the ATLAS detector.

3.1 The LHC Collider

The Large Hadron Collider (LHC) is the largest and highest-energy particle collider ever built. It is a circular synchrotron which lies in a tunnel 27 km long in circumference located at CERN, at a depth varying between 50 and 175 m below the ground. Starting in 2010, the LHC has provided centre-of-mass energies, $\sqrt{s}$, up to 13 TeV (6.5 TeV/beam) and instantaneous luminosities up to few 10^{34} cm$^2 s^{-1}$. The LHC primarily collides proton beams (p-p), but it can also use beams of heavy ions (Pb-Pb). Two beamlines, enclosed in the same cryo-system and magnetic field, house the two magnetically coupled, counter-rotating beams. The radio-frequency cavities keep the particles tightly bundled in bunches of approximately 120 billion units, with a maximum value of 2808 bunches per beam, separated by approximately 7.5 m (or 25 ns), producing about a billion collisions per second [25]. The size of the bunches varies as they circulate around the LHC ring: they are squeezed to a width of about 20 μm at the collision points, where the two beams cross, to increase their density and, therefore, the probability of collision. Several types of superconducting magnets operate at a temperature of 1.9 K, provided by a cryogenic system based on liquid helium. Dipole magnets (for a total of 1232) are used to keep the beams on their circular trajectory, while quadrupole magnets (392) are needed to keep the beams focused, in order to maximize the chances of interaction in the collision points. In these regions, triplet magnets are used to squeeze the beam in the transverse plane, to focus it at the interaction point. In this way, the travelling beam can be significantly larger than it needs to be

at the interaction point, reducing intra-beam interactions.
Before being injected in the LHC ring, protons and heavy ions pass through a chain of smaller accelerators, already built for previous experiments (see Fig. 3.1). Protons are extracted using an electric field, by hydrogen gas ionization, and accelerated to the energy of 50 MeV by the LINAC2, which is the only linear accelerator in the chain. The protons are accelerated to 1.4 GeV by the Proton Synchrotron Booster (PSB) and then they pass through the Proton Synchrotron (PS), where they arrive to 26 GeV. After that, they are transferred to the last accelerator in the chain, the Super Proton Synchrotron (SPS), where they reach the energy of 450 GeV, before to be injected into the two LHC pipelines. The collider has four crossing points, around which are located

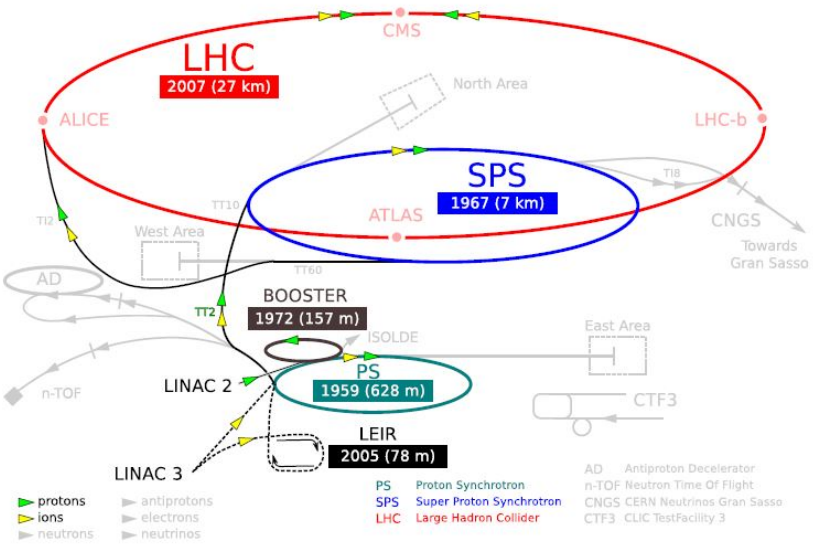

Figure 3.1: Schematic view of the CERN accelerator complex. Only the sections relevant for the LHC operations are highlighted with different colors, while the rest is indicated with a light grey color.

the four major experiments (ATLAS [1], CMS [2], LHCb [26], and ALICE [27]). Three smaller detectors have been installed close to other experiments' caverns: LHCf [28] (near ATLAS), TOTEM [29] (near CMS), and MoEDAL [30] (near LHCb). Each detector was designed for certain kinds of research, including measuring the properties of the Higgs boson and searching for the large family of new particles predicted by supersymmetric theories, as well as other unsolved questions of particle physics. The LHC design specifications allow for collisions in the TeV regime, located just beyond the electroweak symmetry breaking point. At this scale, new physics is expected to manifest itself, according to several theories.

3.2 The ATLAS Detector

ATLAS (A Toroidal LHC ApparatuS) is a multi-purpose detector situated in the experimental cavern at Point 1 along the LHC [1], located approximately 100 m under ground. Thanks to its length of 46 m and diameter of 25 m, this 7000 tonnes particle detector is one of the largest ever constructed. This detector, pictured in Fig. 3.2, is designed to record the collision products across almost the full 4π solid angle, since it is centered around the collision point and extends with cylindrical symmetry around the beam line. Along

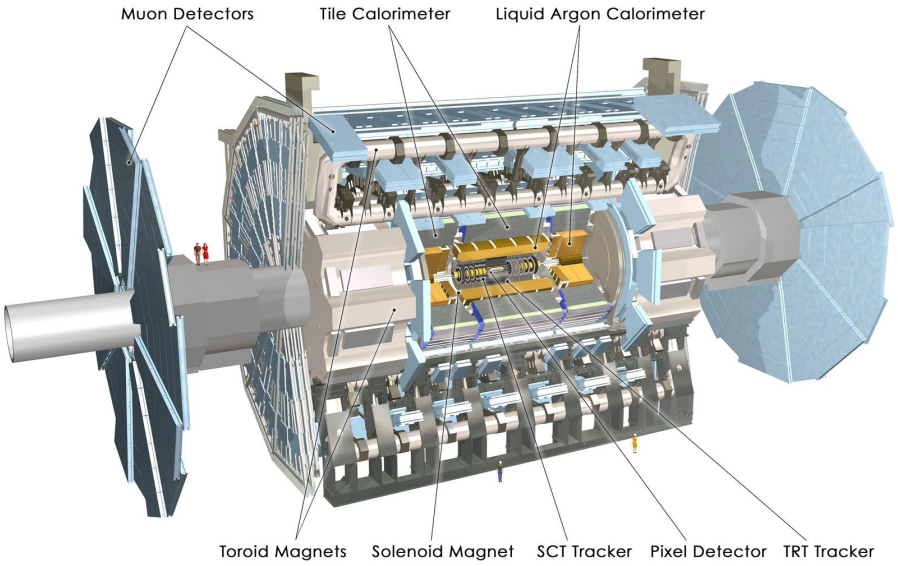

Figure 3.2: The ATLAS detector and its main sub-detectors.

the beam axis, sub-detector layers are arranged as concentric cylinders with endcap regions perpendicular to the beam line. Additional, smaller, forward detectors, such as LUCID and ALFA, contribute to monitoring the luminosity delivered to ATLAS [1]. Each of them plays an important role in the particle reconstruction. Closest to the beam pipe is the Inner Detector (ID), used to reconstruct the trajectory of charged particles. The ID is enclosed by a solenoidal magnet, which provides a strong magnetic field to bend the charged particles and measure their momentum and electric charge. The Electromagnetic (EM) Calorimeter surrounds the ID and is designed to precisely measure the energy of electrons and photons. Outside the EM Calorimeter there is the Hadronic (Had) Calorimeter, which measures the energy of hadronic particles. Finally, the calorimeters are enclosed by the Muon Spectrometer (MS), designed to reconstruct and identify muons, which usually escape the previous detector layers. The MS is embedded in a toroidal magnetic field and consists in tracking chambers, to provide precise measurements of momentum and charge, and detectors used for fast triggering. Each sub-detector is now briefly described.

3.2.1 The Inner Detector

This detector, shown in Fig. 3.3, is designed to trace out the trajectories of charged particles close to the interaction point, with minimal interference on their energy. The particles are deflected and bent into helical motion by a 2 T

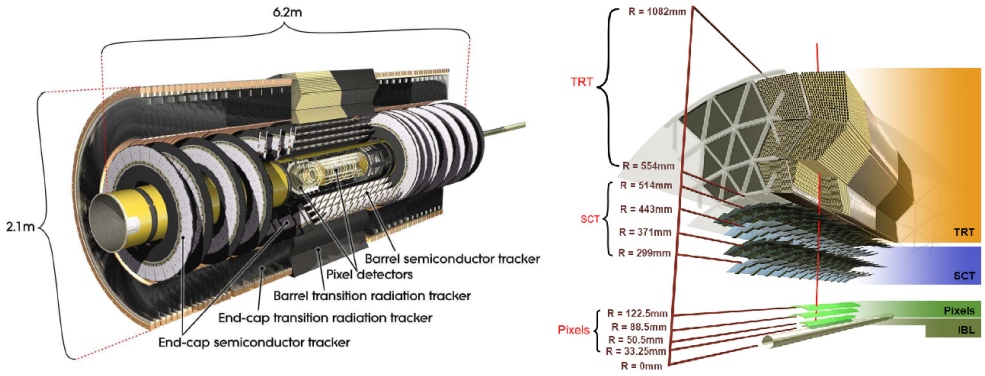

Figure 3.3: Three dimensional diagram of the ATLAS Inner Detector.

magnetic field parallel to the beam line, produced by a superconducting central solenoid magnet that encloses the Inner Detector (ID), and facilitates the measurement of their direction, momentum and charge. Because of its vicinity to the interaction region, the ID requires the highest granularity among all ATLAS sub-detectors. Thanks to the high precision of the track reconstruction in (r, ϕ, z), the ID is able to measure the position of the primary vertex in a collision and to identify secondary vertexes, due to pile-up or in flight decays of unstable particles. As particles travel outward from the collision point, they encounter, in order, the Pixel Detector, the Semi-Conductor Tracker (SCT) and the Transition Radiation Tracker (TRT).

The Pixel Detector

The Pixel Detector (PD) is the closest system to the collision point and it is built directly around the beryllium beam pipe in order to provide the best possible primary and secondary vertex resolution. Its original setup is composed by three cylindrical layers in the barrel region and two end-caps, each consisting of three disks. The PD provides three precision measurement points for tracks with pseudorapidity $|\eta| < 2.5$ and it has a full coverage in ϕ. The basic elements of the PD are the silicon sensor "modules", identical for barrel and disks. The 250 μm thick modules are divided into 50 μm wide and 400 μm long pixels, with 47232 pixels on each of the 1744 modules. The total number of channels for the whole detector is $\sim$ 80.4 millions. In 2014, before Run II, a fourth layer was installed in the barrel region: the Insertable B-Layer (IBL). It adds $\sim 6.02 \cdot 10^6$ pixels, improving the track reconstruction and the vertex measurement. The supporting structure is made of thin carbon-fibre and it is integrated with the cooling system, in order to not perturb significantly the crossing particles, maintain a constant operating temperature of about -7 °C and be radiation-hard ($\sim$800 kGy are expected during the detector life time).

The SCT Detector

The SCT is the second element of the tracking system, going from the beam pipe outwards. It is composed by four cylinders in the barrel region and two end-caps consisting of nine disks. It provides typically eight strip measurements for particles originating in the beam-interaction region. The detector consists of 4088 modules. The strips in the barrel are approximately parallel to the solenoid field and beam axis, and have a constant pitch of 80 μm, while in the end-caps the strip direction is radial and of variable pitch.

The TRT Detector

The TRT is the outermost system of the ID and it consists of 298304 proportional drift tubes (straws), with a read out of $\sim$ 351000 electronic channels. The straws in the barrel region are arranged in three cylindrical layers and 32 ϕ-sectors; they have split anodes and are read out from each side. The straws in the end-cap regions are radially oriented and arranged in 80 wheel-like modular structures. The TRT straw layout is designed so that charged particles with transverse momentum $p_T > 0.5$ GeV and with pseudorapidity $|\eta| < 2.0$ cross typically more than 30 straws. The TRT can also be used for particle identification. Its tubes are interleaved with layers of polypropylene fibres and foils: a charged particle passing through the boundary region between materials with a different refraction index emits X-ray radiation whose intensity is proportional to the relativistic factor. The TRT works with two threshold levels, defined at the level of the discriminator in the radiation-hard front-end electronics.

The Cooling System

For the Pixel Detector and the SCT, cooling is necessary to reduce the effect of the radiation damage on the silicon. They share a cooling system, which uses C_3F_8 fluid as a coolant. The target temperature for the silicon sensors after irradiation is 0°C for the Pixel Detector and -7° C for the SCT. Because the TRT operates at room temperature, a set of insulators and heaters isolates the silicon detectors from the ATLAS environment.

The whole ID system is embedded in a 2 T solenoidal magnet. The flux is returned by the steel of the ATLAS hadronic calorimeter and its girder structure. As a result there is a negligible field within the EM Calorimeter volume and a small field in the Had Calorimeter volume.

3.2.2 The Calorimeters

The calorimeter system includes the EM (for electrons and photons) and the Had Calorimeters. The calorimeter system is hermetic out to $|\eta| < 4.9$ and it is ~ 9 - 13 λ thick, enough to capture the 99% of the hadronic showers from single charged pions up to $\sim$ 500 GeV. A schematic view of the calorimeter system is shown in Fig. 3.4. The main purpose of the calorimeters is to measure

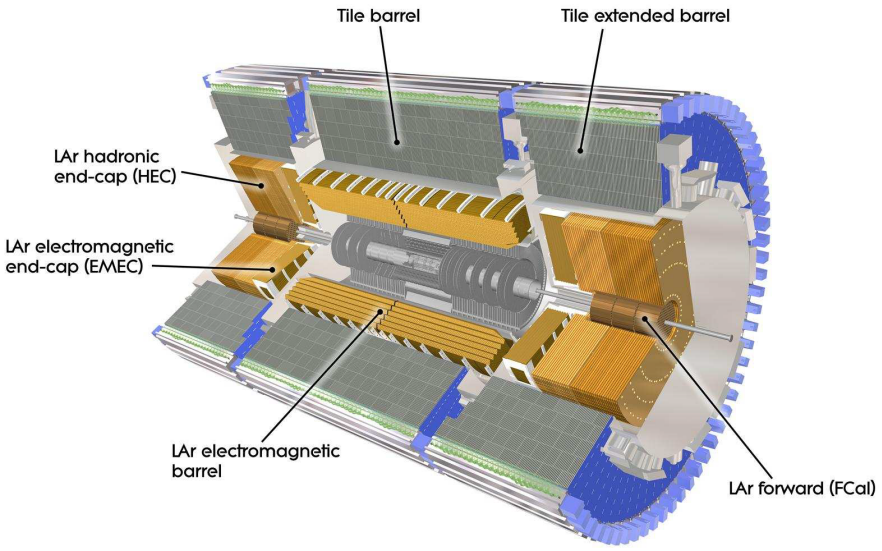

Figure 3.4: Schematic view of the ATLAS calorimeter system.

the energy of the particles and their position. One of the most important requirements is to provide good containment for electromagnetic and hadronic showers, as well as limit the punch-through into the muon system. Therefore, the total thickness of the EM calorimeter is more than 22 X_0 in the barrel and more than 24 X_0 in the End-Caps. It contains electron and photon showers up to $\sim$ 1 TeV and it also absorbs almost 2/3 of a typical hadronic shower. The approximate 9.7 (10) interaction lengths (λ) of the active EM + Had calorimeter in the Barrel (End-Caps) are adequate to provide good resolution for high-energy jets. Some details on the different calorimeter regions are given below.

The Electromagnetic Calorimeters

The EM calorimeter is a sampling calorimeter made of lead and Liquid-Argon (LAr), divided into one Barrel part ($|\eta| < 1.375$) and two End-Caps ($1.375 < |\eta| < 3.2$), each one with its own cryostat. The Barrel calorimeter consists of two identical half-barrels, separated at $z = 0$. Each End-Cap is mechanically divided into two coaxial wheels: an inner wheel covering the region $1.375 < |\eta| < 2.5$, and an outer wheel covering the region $2.5 < |\eta| < 3.2$. Over the region devoted to precision physics ($|\eta| < 2.5$), the EM calorimeter is segmented into three longitudinal parts: the strips, middle and back sections.

While most of the electron and photon energy is collected in the middle, the fine granularity of the strips is necessary to improve the $\gamma - \pi^0$ discrimination and the back measures the tails of highly energetic electromagnetic showers, and helps to distinguish electromagnetic and hadronic deposits. For the End-Cap inner wheel, the calorimeter is segmented in two longitudinal sections and has a coarser lateral granularity than for the rest of the acceptance. Since most of the central calorimetry sits behind the cryostat, the Solenoid, and the 1-4 λ thick ID, EM showers begin to develop well before they are measured in the calorimeter. In order to take into account and correct for these losses, up to $|\eta| = 1.8$ an additional presampler layer is mounted in front of the sampling portion of the calorimetry, which includes fine segmentation in η. The transition region between the Barrel and End-Cap EM calorimeters, $1.37 < |\eta| < 1.52$, is expected to have a poorer performance because of the higher amount of passive material in front; this region is often referred to as "crack region".

The Hadronic Calorimeters

The Had Calorimeter is realized with a variety of techniques depending on the region: Central, End-Cap and Forward. In the central region there is the Tile Calorimeter (Tile), which is placed directly outside the EM calorimeter envelope. The Tile is a sampling calorimeter which uses steel as absorber and scintillating tiles as active material. It is divided into a Barrel ($|\eta| < 1.0$) and two Extended Barrels ($0.8 < |\eta| < 1.7$), longitudinally segmented in three layers. The Hadronic End-Cap Calorimeter (HEC) consists of two independent wheels per end-cap, located directly behind the End-Cap EM calorimeter and sharing the same LAr cryostats. It covers the region $1.5 < |\eta| < 3.1$, overlapping both with the Tiles and the Forward Calorimeter. Each wheel is divided into two longitudinal segments, for a total of four layers per End-Cap. The wheels are made by Copper plates are interleaved with 8.5 mm LAr gaps, providing the active medium for this sampling calorimeter.

The Forward Calorimeter

The Forward Calorimeter (FCal) covers the $3.1 < |\eta| < 4.9$ region and is another LAr based detector. Integrated into the End-Cap cryostats, it consists of three thick independent modules in each End-Cap: the absorber of the first module is Copper, which is optimised for electromagnetic measurements, while for the other two is Tungsten, which is used to measure predominantly the energy of hadronic interactions. Both materials have been chosen for their resistance to radiation. The region where the FCal is set, is very close to the beam pipe, so that the expected radiation dose is very high. Therefore the electrode structure is different from the accordion geometry, consisting in a structure of concentric rods and tubes parallel to the beam axis. The LAr in the gap between the rod and the tube is the sensitive medium.

3.2.3 The Muon Spectrometer

The muon system has two different functions: it is needed for high precision tracking of muons and also for triggering on them. The layout of the MS is shown in Fig. 3.5. The momentum measurement is based on the reconstruction

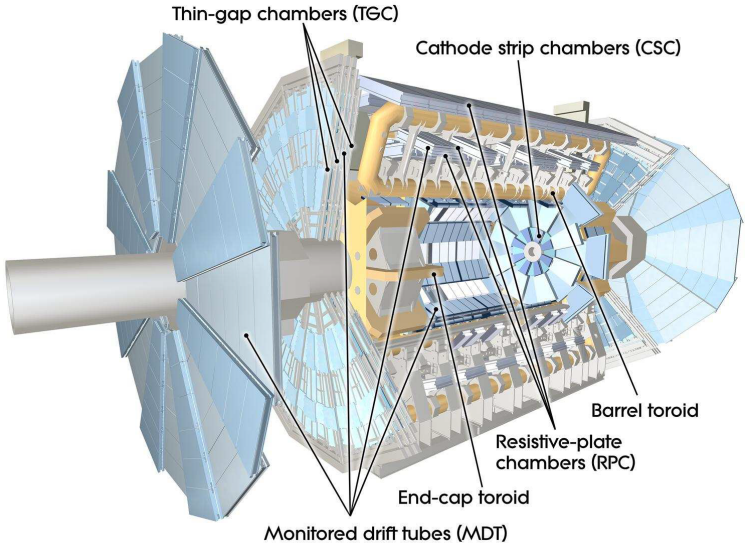

Figure 3.5: Schematic view of the various MS components.

of the muon's trajectories bent by a magnetic field. The large volume magnetic field is provided by the Barrel toroid in the region $|\eta| < 1.4$, by two smaller End-Cap magnets in the $1.6 < |\eta| < 2.7$ region and by a combination of the two in the transition region ($1.4 < |\eta| < 1.6$). This magnet configuration provides a field mostly orthogonal to the muon trajectories, centered on the beam axis, perpendicular to the solenoidal field that serves the ID. Since the toroidal magnet system of the MS is completely independent of the Solenoid in the ID, ATLAS is able to acquire two independent measurements of a muon momentum. The momentum measurement is performed over most of the η-range by the Monitored Drift Tubes (MDT). At large η and close to the interaction point, Cathode Strip Chambers (CSC) with higher granularity are used: they have been designed to withstand the demanding rate and background conditions. Concerning the triggering function of the muon system, it covers the pseudorapidity range $|\eta| < 2.4$. Resistive Plate Chambers (RPC) are used in the barrel and Thin Gap Chambers (TGC) in the end-cap regions. The trigger chambers for the MS serve a threefold purpose: to provide bunch-crossing identification, to provide well-defined transverse momentum thresholds and to measure the muon coordinate in the direction orthogonal to that determined by the precision-tracking chambers. The barrel chambers are positioned on three cylinders concentric with the beam axis. They cover the pseudorapidity range $|\eta| < 1$. The end-cap chambers cover the range $1 < |\eta| < 2.4$ and are arranged in four disks concentric with the beam axis. At $|\eta| = 0$ there is a gap in the detector for cable routing.

3.2.4 Luminosity Detectors

One measurement which is very important for almost every physics analysis is the luminosity measurement. As it is a fundamental quantity, three different detectors help in its determination. At ±17 m from the interaction region there is the LUCID (LUminosity measurement using Cerenkov Integrating Detector). It detects inelastic *pp* scattering in the forward direction and it is the main online relative-luminosity monitor for ATLAS. It is also used, before collisions are delivered by the LHC, to check the beam losses. For the beam monitoring, another detector has been inserted: the BCM (Beam condition Monitor). The other detector used for luminosity measurement is ALFA (Absolute Luminosity For ATLAS). It is located at ±240 m from the interaction point. It consists of scintillating fibre trackers located inside Roman pots which are designed to approach as close as 1 mm from the beam. The last detector is ZDC (Zero-Degree Calorimeter). It is located just beyond the point where the common straight-section vacuum pipe divides back into two independent beam-pipes. Neutral particles with $|\eta| \geq 8.2$, not being affected by the magnetic fields which bend the proton beams, are detected and measured by the ZDC modules, consisting of layers of alternating quartz rods and tungsten plates.

3.3 Particle Identification and Reconstruction at ATLAS

The ATLAS detector is able to identify and measure the properties of most of the SM particles that can be produced by the high-energy *p*-*p* collisions produced by the LHC. The information from the different sub-detectors described in Section 3.2 are combined together in order to distinguish and reconstruct highly energetic electrons, muons, photons and hadrons. Electrons and photons are mainly identified through the information from the EM Calorimeter, muons through the MS and hadrons mostly via the Had Calorimeter, with the ID information contributing to all of these identification and measurement processes. A particle traveling through the ATLAS detector with enough energy generates different types of signals in the various sub-detectors. Charged particles leave *tracks* (reconstructed by connecting a number of precise position measurements) when passing trough the layers of the ID and the MS, and since both sub-detectors are immersed in magnetic fields, the curvature of these tracks is used to both identify the sign of their electric charge and to measure their momentum (by assuming unitary electric charge). Particles interacting either electromagnetically (i.e. charged particles and photons) or strongly (hadrons) interact with the EM and Had Colorimeters, and are typically completely stopped before reaching the MS. The only particles reaching the MS are muons, due to their low energy loss in the calorimeters, and neutrinos. Electrons are reconstructed from *clusters* of energy deposits in the EM Calorimeters associated to reconstructed tracks in the ID. The algorithms for electron reconstruction (based on multi-variate discriminants) are opti-

mised for the discrimination between signal isolated electrons and background electrons from photon conversion, object misidentification and hadron decays. Photons are reconstructed in a similar way as electrons, but requiring the absence of any matched ID track. Muons are reconstructed combining the reconstructed tracks by the ID and the MS, and are discriminated from background sources by sets of requirements on the quality and compatibility of these tracks. Jets of hadrons, which arise from the production of high energy quarks or gluons in the high-energy collisions as described in Section 2.1.2, are characterised by multiple energy deposits in the calorimeters and by the presence of several tracks in the ID. In ATLAS the jet reconstruction starts from grouping together energy deposits in adjacent cells of the calorimeters, and proceeds by clustering them with the so-called anti-k_t algorithm [31]. Tracking information from the ID is then combined to the calorimeter information to both improve the energy and direction measurement and to provide b-tagging information. This b-tagging procedure is used to identify jets coming from the fragmentation of b-quarks, and makes use distinctive features of the b-hadron decays such as their relatively long lifetime and their large mass, combined together through machine learning algorithms providing optimal performance in terms of efficiency and purity.
The ATLAS detector is not able to detect and measure neutrinos. However, an unbalance between the initial and final transverse momentum can be an indirect indication of the presence of one or more escaping neutrinos, being the only SM particles not being detectable by ATLAS. This unbalance is referred to as the missing energy in the transverse plane, usually denoted as E_T^{miss}, MET or $\not{E}_T$ and given by:

$$(E_T^{miss})^2 = (E_x^{miss})^2 + (E_y^{miss})^2 \tag{3.1}$$

The direction of E_T^{miss} in the transverse plane is obtained from the azimutal angle ϕ^{miss}:

$$\phi^{miss} = \tan^{-1}\left(\frac{E_y^{miss}}{E_x^{miss}}\right) \tag{3.2}$$

3.3.1 Coordinate System and Conventions

The ATLAS right-handed coordinate system, used to describe position within the detector volume, is defined as follows. With the origin located at the designated collision point, the z direction extends along the beam-line, the positive y direction points upwards towards the sky, and the positive x direction points towards the center of the LHC ring. The x and y axes form the so-called transverse plane, perpendicular to the beam line, and vector components in this plane are called transverse components. In particular, the transverse momentum p_T is defined as:

$$p_T = \sqrt{p_x^2 + p_y^2} = p\sin(\theta) \tag{3.3}$$

Observables in the transverse plane play a critical role in ATLAS analyses because they are invariant to unknown boosts along the beam axis due to

initial velocities of the colliding partons with respect to the center-of-mass frame. Because of the cylindrical symmetry in the detector geometry, it is more suitable to define angular coordinates: ϕ, the azimuthal angle, revolves around the beam direction in a counter-clockwise direction from the x axis upwards, while θ, the polar angle, is the angle that defines the aperture with respect to the positive beam axis. The angle ϕ is zero towards the center of the ring, while θ is zero along the beam line. The polar angle is reparameterized in terms of the more commonly used pseudorapidity:

$$\eta = -\ln\tan(\theta/2) \tag{3.4}$$

Particles traveling along the beam direction (forward particles) have $\eta = \infty$, and particles perpendicular to it (central particles) have $\eta = 0$. The usefulness of the definition of the η coordinate relies on the invariance of distances in η under unknown boosts along the beam direction, which results in a nearly uniform particle distribution in η. The angular distance between two particles produced in the ATLAS detector is given by ΔR:

$$\Delta R = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2} \tag{3.5}$$

with $\Delta\eta = \eta_2 - \eta_1$ and $\Delta\phi = \phi_2 - \phi_1$.

Chapter 4

Deep Learning

The data collected by the LHC experiments are vast in both the number of collisions and in the complexity of each collision. The traditional way to analyze the data (or generate simulated data) is to first develop algorithms based on domain knowledge, then implement them in software, and use the resulting programs to analyze (or generate) data. This process is labor intensive, and analyzing complex datasets with many input variables becomes increasingly difficult and sometimes intractable. Machine Learning (ML) and the subfield of Deep Learning (DL) attack these problems in a different way: instead of humans developing highly specialized algorithms, computers learn from data how to analyze complex data and produce the desired results, with relatively small human intervention. In this chapter the basic concepts of ML are first provided in Section 4.1. Among the various ML methods, that of artificial neural networks is treated with particular attention in Section 4.2. By structuring the chapter in this way, it is possible to get step by step to deep learning, which is discussed in Section 4.3 as a progressive development of previous techniques.

4.1 Machine Learning

Machine Learning (ML) is a class of algorithms that can make predictions without being explicitly programmed to perform a specific task. These programs produce the desired results following a "learning from data" approach, by directly looking at a dataset to improve their own performance, often with few or no assumptions on the underlying processes. In ML, basically two different approaches are used:

- Supervised learning: solved examples are shown to the algorithm, which looks at the desired output to optimize his own output.
- Unsupervised learning: information are extracted from data without providing any solutions, because inaccessible or unimportant for the task. Data are grouped according to the similarities that are found following clustering methods.

Unsupervised learning algorithms are often used to pre-process the data before applying supervised learning models. There are some techniques that

lie in between these two main categories, such as semi-supervised, active and reinforcement learning. Considered that the technique applied in this thesis is based on a supervised learning approach, more details about this class of algorithms are given in the following dedicated section.

4.1.1 Supervised Learning

The algorithms based on a supervised learning approach are mathematical models that contain parameters. Usually, several thousand parameters are used, depending on the complexity of the task. The way the model architecture is defined is quite general and it is the particular set of parameters that allows to solve a specific task. The learning procedure, which is used to tune such parameters, is applied in a phase called *training*. The solved examples that are here provided contain input-output pairs. The quantity of interest that you want to predict is named *target*, but it is usually called *label* when it is provided as a correct solution. The aim of the training is learning how to generalize, i.e. to make accurate predictions on new unseen data, that have characteristics similar to the training set, but whose labels are unknown. Once the training has been completed, the model testing is performed at a later phase, that is often referred to as *application*, when the program makes predictions on a new test dataset. In the training set, the inputs are provided in the form of a matrix $\boldsymbol{X}$, made by N rows of independent and identically distributed observations $\boldsymbol{x}^{(i)}$. The index N corresponds therefore the dataset size, i.e. how many different examples are provided to the algorithm. Each observation is a vector consisting of m components of variables or proprieties, called *features*, representing all the information needed to solve the task. In high energy physics (HEP), an observation is typically a collision event and the features are the variables measured by detectors, or any other relevant quantity. Each $\boldsymbol{x}^{(i)}$ is coupled to a label $\boldsymbol{y}^{(i)}$, which can be either a scalar or a vector with a certain dimension l, that codify some quantity of interest and forms, along with all the examples, the matrix of targets $\boldsymbol{Y}$. The two matrices that form the training set are schematized in Eq. 4.1.

$$\boldsymbol{X} = \begin{bmatrix} x_1^{(1)} & \dots & x_m^{(1)} \\ \vdots & \ddots & \\ x_1^{(N)} & & x_m^{(N)} \end{bmatrix}, \quad \boldsymbol{Y} = \begin{bmatrix} y_1^{(1)} & \dots & y_l^{(1)} \\ \vdots & \ddots & \\ y_1^{(N)} & & y_l^{(N)} \end{bmatrix} \tag{4.1}$$

The training and testing datasets share the same number of features, being the only difference between them the presence or absence of the label. The number of solved examples, required to successfully perform the training, strongly depends on the specific problem and can be up to many millions of events, while the number of rows in the test set can be any value. A supervised learning problem can be formulated in terms of a search for some function f, called *predictor*, that maps input features to the lower dimensional space of the desired targets $\boldsymbol{Y}$ (Eq. 4.2).

$$f : \boldsymbol{X} \to \boldsymbol{Y} \tag{4.2}$$

During the training phase, the algorithm optimizes some metric $\mathcal{L}(f(\boldsymbol{X}), \boldsymbol{Y})$ of our choosing. This function, usually named *loss function*, is calculated over the entire dataset points and measures how good is the prediction $f(\boldsymbol{X})$ with respect to the desired output $\boldsymbol{Y}$. There are two major types of problems that can be solved by supervised ML algorithms, namely *classification* and *regression* tasks. In the first case, the goal is to group events in two (*binary classification*) or more (*multi-class classification*) classes. Popular examples in HEP are signal/background discrimination and particle identification. Integer numbers are used as labels to mark each class of events, being a choice from a predefined list of possibilities, and some definition of accuracy is maximized. Regression tasks differ from classification tasks in the nature of the output they attempt to predict. The goal here is to approximate one or more real valued quantity of interest, whose domain may be bounded or unbounded. Being the output generally continuous, real numbers are used as labels, or floating point numbers in programming terms. In a regression task, examples of common choices of loss functions are the Mean Absolute Error (MAE) (Eq. 4.3) or the Mean Squared Error (Eq. 4.4):

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_{true} - y_{pred}| \tag{4.3}$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_{true} - y_{pred})^2 \tag{4.4}$$

In these functions, the absolute or squared difference between the true and the predicted output is averaged over the entire training dataset. Both the codomains are made up of non-negative numbers and this fact ensures that small values for such metrics correspond to an optimal weight configuration, being 0 the ideal value. If, instead, both positive and negative numbers enter in the sum, a null value does not necessarily represent the optimal combination. The procedure of the $\mathcal{L}$ optimization is quite general and many different algorithms belong to supervised ML models. While an in-depth discussion of each class of algorithms is beyond the scope of this thesis, *artificial neural networks* are treated in detail in the next sections. Some other popular methods in ML are *k-nearest neighbour*, *decision trees* and *linear models*.

4.2 Neural Networks

Neural Networks (NNs) are a class of various computing systems loosely inspired by the human brain architecture, where, making an extreme simplification, information are codified and transmitted by electric signals through a complex network made of a basic unit, the neuron. In these biological networks, each neuron receives input signals from other $\simeq 10^3$ cells, then it processes this information and transmits an output (*spike signal*) through the synapses, but only if the stimulus is above a threshold potential. One could mathematically

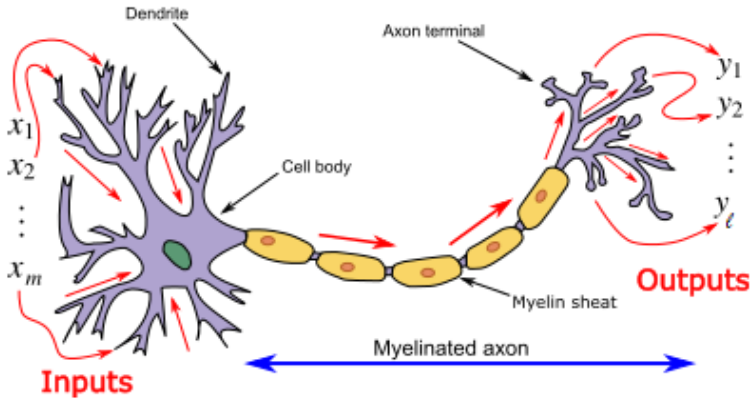

Figure 4.1: Schematic representation of a biological neuron that receives n input signals and produces m outputs.

describe this behaviour through the following simple model, proposed by McCulloch and Pitts in 1943 [32]. Each neuron receives m inputs x_i, each of them weighted by a parameter w_i. The neuron is firing when the weighted sum of the inputs is above a threshold w_0. This model can be mathematically represented by the Eq. 4.5:

$$\hat{y} = \Theta(\sum_{i=1}^{m} x_i w_i - w_0) \tag{4.5}$$

The Heaviside step function $\Theta(z)$ (0 if $z < 0$ and 1 if $z \geq 0$) is applied to reproduce the two possible states of the neuron, namely at rest or firing. This model was first practically implemented by Rosenblatt in 1958 [33], with the construction the *perceptron*, intended to be a machine rather than a program. The perceptron is the simplest model for an artificial neuron and can work only as a binary classifier. The use of a simple step function in Eq. 4.5, motivated by the biological inspiration, allows to effectively build logic circuits, since it can compute fundamental operations as AND, OR, NAND, but it also shows some limits when it is used to learn a complex task or to perform a regression. In general, a learning algorithm works properly if small changes in the model lead to small changes in the output. In a perceptron, small variation in one of the parameter can completely overturn the result, since the smallest change in the output is actually the only one possible.

To solve the issue of the discreteness of the output, it is useful to replace $\Theta(z)$ with an appropriate non-linear *activation function* $g(z)$. By introducing the input and weight vectors, a compact notation can be obtained (Eq. 4.6). in the following, the quantity z is used to denote the linear combination of the inputs or, more generally, the argument of the activation function.

$$\hat{y} = g(\mathbf{X}^T\mathbf{W} + w_0) = g(z) \tag{4.6}$$

A schematic representation of such artificial neuron is shown in Fig. 4.2. There

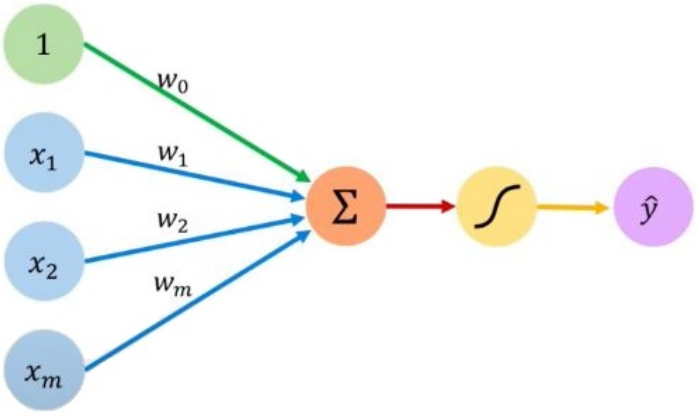

Figure 4.2: Schematic representation of a single output artificial neuron. All of the m features of an event are multiplied by a corresponding weight w_i and summed together with a bias w_0. The obtained quantity is passed to the non-linearity to produce the final output $\hat{y}$.

are many possible choices for the $g(z)$ function, that are made empirically. Some of the most popular choices are discussed below.

4.2.1 Activation Functions

Depending on the application task, the target's domain may be bounded or unbounded. It is important to make an appropriate choice of activation function, in order to produce an output that falls within the physical range observed over the training samples. Moreover, the purpose of activation functions is also to introduce non-linearities into the network. When the activation function is non-linear, then a particular arrangement of neurons, consisting of multiple neuron layers, can be proven to be a universal function approximator [34], allowing to approximate arbitrarily complex functions [1]. Linear activation functions cannot satisfy this property, because even if multiple neurons are combined in arbitrarily complex architectures, the composition of such functions will also result in a linearity, with inadequate performance for most of the practical cases of application, no matter how deep or large the NN is.

[1] This important result is known as the Universal Approximation Theorem [34].

Many choices of non-linear activation function are possible, beyond the already mentioned step function. They all share the fact that they have a simple expression for the first derivative, which is useful in applying optimization algorithms, as will be discussed below. Until the past decade, the most commonly used activation functions in neural networks were the sigmoid (Eq. 4.7) and the hyperbolic tangent (Eq. 4.8):

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{4.7}$$

$$tanh(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}} \tag{4.8}$$

These are monotonically increasing functions with a similar asymptotic behavior (Fig. 4.3). While the sigmoid is bound between 0 and 1, the hyperbolic tangent is bound between -1 and 1 and is symmetric with respect to the origin. These two functions are still sometimes used for classification purpose, being the output finite, and because of the simple expression of their first derivative (Eqs. 4.9 and 4.10).

$$\sigma'(z) = \sigma(z)(1 - \sigma(z)) \tag{4.9}$$

$$tanh'(z) = 1 - tanh(z)^2 \tag{4.10}$$

Nowadays, their usage has faded due to their issues with saturation. In both equations 4.7 and 4.8, as $|z| \to \infty$, $g'(z) \to 0$ which, as explained in Sec. 5.3, can lead to rapidly vanishing gradients [35]. A more commonly used activation function, that does not suffer from such issues, is the Rectified Linear Unit (ReLU), which is linear if its argument is positive, and null if it is negative:

$$g(z) = \begin{cases} x, & x > 0 \\ 0, & x \leq 0 \end{cases} \tag{4.11}$$

$$g'(z) = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases} \tag{4.12}$$

A comparison between these three functions is shown in Fig. 4.3. In addition to added benefits from lack of saturation in the output, ReLU offers computational speedups, as their gradients can only ever be 0 or 1, and can be computed via a single comparison. It was demonstrated that this function enables to train deep supervised neural networks without requiring unsupervised pre-training [36]. In this case, a faster and more effective training is also achieved on large and complex datasets, compared to other widely used activation functions [36]. The usefulness of such functions becomes evident when, instead of using a single neuron, such fundamental building blocks are combined together to form a *neural network* (NN), where the predictor f in Eq. 4.2 is parameterized by a set of hyperparameters θ, made by all weights and biases in the model, which are tuned during the training.

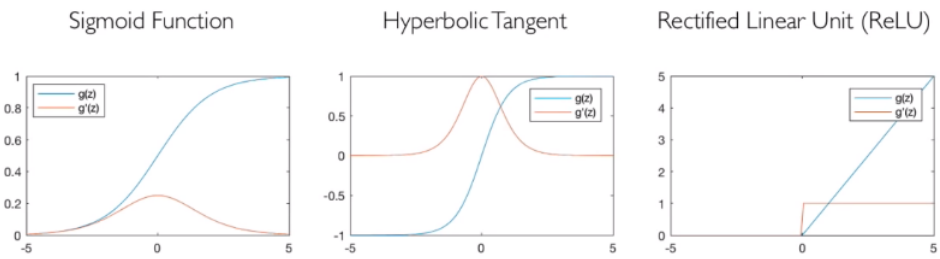

Figure 4.3: Three popular non-linear activation functions with their first derivatives.

4.2.2 Neural Network Architectures

A neural network is based on the collective effect of multiple processing units, the neurons, that are simple functions. The neurons that compose a network are often called *nodes* and can be arranged in several ways. The amount of neurons, the types of connection among them and the choice of the activation function are collectively referred to as the architecture of the network. The neurons are typically organized into layers. Neurons belonging to the same layer receive the same inputs and apply the same activation function, but each one computes a different linear combination z_i sum by using a different set of weights. The layer that receives external data is called input layer, while the final results are produced in the output layer. Two adjacent layers can be connected in several ways. Artificial neural networks can be therefore categorized in two main different groups [37], according to the connection pattern, as schematically illustrated in Fig. 4.4.

- Feed-forward NNs: the information flows forward from the input layer to the output layer, passing through any other internal neuron layer, if present, without cycles.

- Recurrent NNs: The connection between neuron nodes forms a circle. The information is propagated forward but also backwards, from later processing stages to earlier stages.

Other approaches are possible and many different architectures belong to these two classes. The simplest example of ANN is the feed-forward single-layer perceptron, which only consists of an input and an output layer. The single output perceptron, already described in Fig. 4.2, is the simplest case of this class, but also multi-output networks are possible, where each output node consists of a perceptron (Fig. 4.5). The inputs are fed directly to the output nodes following the rule already described in Eq. 4.6. The word perceptron continues to be used for historical reasons, although, strictly speaking, these models are not actually composed of perceptrons, since the activation function is not generally a step function. Single-layer perceptrons are very limited due to the small number of connections and the lack of hierarchy in the characteristics that the network can extract from the data. This means that these models are able to combine the incoming data only once and are not capable of learning high

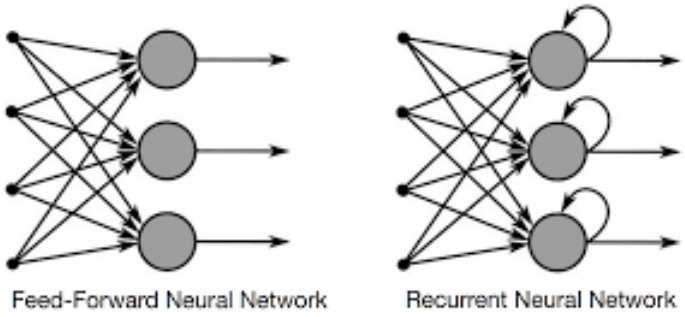

Figure 4.4: Comparison between feed-forward and recurrent neural network architectures. Here, each dot on the left is an input data feature, circular nodes are artificial neurons and the arrows represent the connections between nodes.

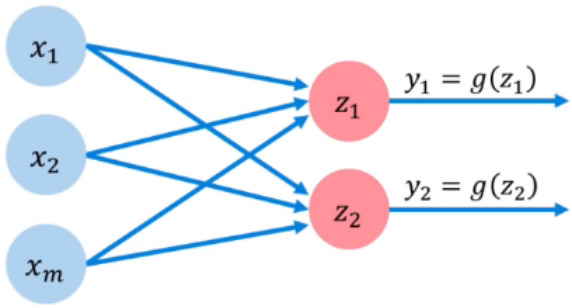

Figure 4.5: An example of feed-forward single-layer perceptron with two outputs. The neurons calculate a different weighted sum, starting from the same inputs, and apply the same activation function to produce the outputs.

level patterns. A paper published in 1969 showed that it was impossible for such networks to learn an XOR function [38]. An alternative way to structure the network is to have multiple layers with fewer neurons. As already mentioned, such multi-layer perceptrons can satisfy the Universal Approximation Theorem. These architectures tend to outperform single layer networks, even if the same total number of parameters is used [39].

4.2.3 Multi-Layer Perceptron

In this architecture, there are a certain number of neuron layers between the input and the output layers. They are called *hidden* layers because, unlike the inputs and the outputs, which are provided or read by the user, its states are not directly observable, representing intermediate processing steps. In this section, the case of *dense* layers is considered, where all the neurons of one layer are fully connected to all nodes of the immediately preceding and immediately following layers. This architecture is not the only one possible, but it is the most common. A formally simple case to describe is that of single hidden

layer NN, shown in Fig. 4.6. Since there are two transformations, the first

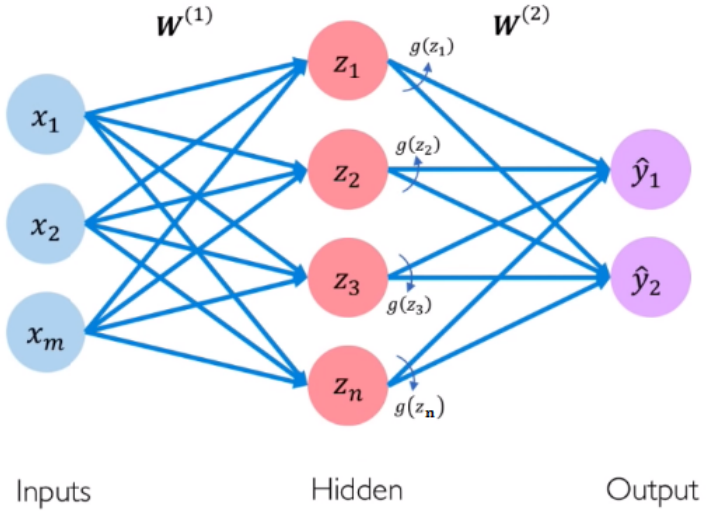

Figure 4.6: Schematic representation of a two-layer neural network, with two outputs. The two layers of neurons are arranged in a single hidden layer and in an output layer. In each hidden node, a different weighted sum z_i of the inputs is calculated through the matrix $\boldsymbol{W}^{(1)}$. The results pass through a non-linearity and two weighted sums are calculated in output using the matrix $\boldsymbol{W}^{(2)}$.

one between the inputs and the hidden layer, and the second one between the hidden layer and the output layer, two weights matrices $\boldsymbol{W}^{(1)}$ and $\boldsymbol{W}^{(2)}$ are used. The first one acts on the input features, as described in Eq. 4.13, to produce the weighted sum z_i for the i-th neuron.

$$z_i = w_{0,i}^{(1)} + \sum_{j=1}^{m} x_j w_{j,i}^{(1)} = w_{0,i}^{(1)} + \left(\mathbf{X}^T \mathbf{W}^{(1)}\right)_i \tag{4.13}$$

In a second step, the non-linearity is applied to each term z_i. The quantity $g(z_i)$ so obtained, i.e. the output of each hidden neuron, now becomes an input for the final layer. Similarly, the second matrix $\boldsymbol{W}^{(2)}$ is used to compute a weighted sum in each of the multi-output node. Another activation function $g^{(2)}$, in general different from the previous one, is then applied to compute $\hat{y}_i$.

$$\hat{y}_i = g^{(2)}\left(w_{0,i}^{(2)} + \sum_{j=1}^{n} g(z_j) w_{j,i}^{(2)}\right) \tag{4.14}$$

This model can be even more improved by simply increasing the number of hidden layers, giving rise to the deep learning architecture. It is useful to introduce a compact notation to denote the fully connected layers, before to describe deep neural networks. Fig. 4.6 can be redrawn as in Fig. 4.7, where an appropriate symbol denotes the dense connection between layers.

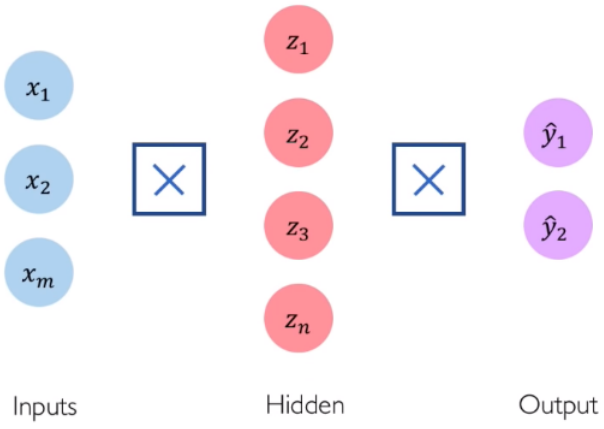

Figure 4.7: A two-layer neural network with two outputs, equivalent to Fig. 4.6, drawn with a compact notation.

4.3 Deep Neural Networks

The emergence of deep learning began around 2012, when a convergence of techniques enabled training of very large neural networks that greatly outperformed the previous state of the art. This allowed to handle higher dimensional and more complex problems than previously feasible. Such MLPs perform multiple stages of processing before to make a prediction. Having large neural networks, made up of many hidden layers, is what inspired the term *deep learning*. The power of modern deep neural networks (DNNs), compared to their shallow predecessors, is largely to be attributed to the benefits that arise in connection to their depth. The hierarchical nature of the model allows to build nested representations of the data, enhancing predictive features and suppressing less relevant information. In many applications of ML methods, it is not possible to directly use raw data (e.g. the variables measured by a detector) as a training dataset. A prior step of feature engineering is needed, where a human intelligence must devise variables that are sensible for the specific case of application, combining the dataset features to create an abstract representation of the raw data. Deep learning is able to learn such high levels of abstraction on its own [40]. A schematic representation of a DNN is shown in Fig. 4.8. Some DNN models often have dozens of layers and can contain up to hundreds of millions of parameters, optimized with a similar amount of training examples. In the last decades, DNNs have been widely and successfully used in many fields, due to the advent of big data, improvements in both hardware and software technologies and also due to theoretical breakthroughs. Some of the most prominent examples of successful applications are the recognition of images [41], the speech-to-text synthesis [42], the generation of human faces [43] or the prediction of new drugs [44]. A relevant example in high energy physics is the application of DL in the context of exotic particle searches at the LHC [45]. This study showed the capability of improving the classification of signal and background events in two benchmark scenarios

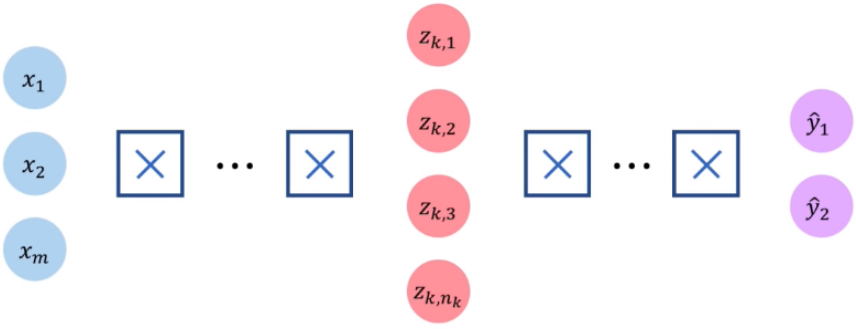

Figure 4.8: Schematic representation of a deep neural network with multiple hidden layers and two outputs. An appropriate symbol was introduced to indicate the dense connection between layers.

(new heavy Higgs bosons and SUSY particles) with the use of DNNs and raw data, with respect to more traditional shallow techniques that use engineered high-level features.

4.3.1 The DL Problem

The learning problem initially presented in Section 4.1.1 can be now reformulated in terms of a deep predictor $\hat{Y}(X)$, that is a multivariate function constructed by L blocks of hidden layers. Given $g^{(1)}$, ..., $g^{(L)}$ non-linear activation functions, the activation rule for each layer l is given by:

$$g^{(l)}(z) = g^{(l)}(W^{(l)}z + w_0^{(l)}) \tag{4.15}$$

The deep predictor is defined as a composite map:

$$\hat{Y}(X) = (g^{(l)} \circ ... \circ g^{(L)})(X) \tag{4.16}$$

The deep predictor can also be defined as a computation graph, where the i-th node in the l-th layer is given by:

$$a^{(0)} = X \tag{4.17}$$

$$z_i^{(l)} = w_{0,i}^{(l)} + \sum_{j=1}^{N_l} w_{i,j}^{(l)} a_j^{(l-1)} \tag{4.18}$$

$$a_i^{(l)} = \left(g^{(l)}(a^{(l-1)})\right)_i = g^{(l)}\left(z_i^{(l)}\right) \tag{4.19}$$

$$\hat{Y}(X) = a^{(l)} \tag{4.20}$$

A class of methods widely used to solve such supervised learning problem is presented in Chapter 5.

4.3.2 DNN Architectures

Deep neural networks themselves contain several different models. Since the case of recurrent neural networks has already been treated, other two popular architectures are briefly described below.

Convolutional Neural Networks (CNNs)

ConvNets are used to process data in the form of multiple arrays, in particular digital images (Fig. 4.9). They perform filtering on the input data in order to create a feature map of relevant characteristics for the task, reducing the dimensionality of the data. CNNs use relatively little pre-processing compared to other image classification algorithms. This means that the network learns the filters that in traditional algorithms were hand-engineered. ConvNets may include local or global pooling layers to reduce the dimensions of the data by combining the outputs of neuron clusters at one layer into a single neuron in the next layer. Local pooling combines small clusters, typically 2 x 2. Global pooling acts on all the neurons of the convolutional layer [46]. Pooling and convolutional layers are often used at the first stages of the network, just after the input layer, along with a non-linear activation. Dense layers follow the same structure of a fully-connected MLP, which gives the final output in the last layer.

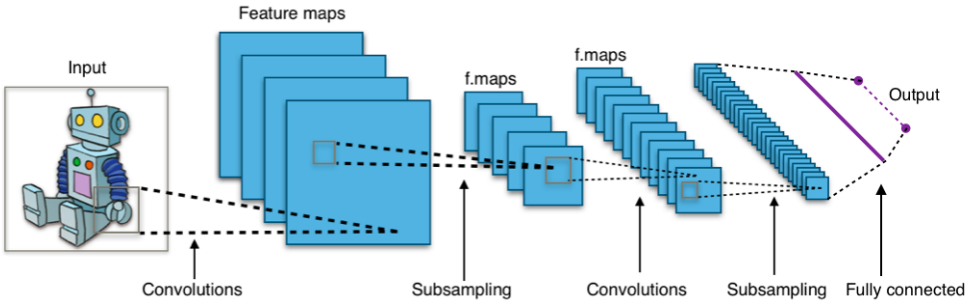

Figure 4.9: A Convolutional Neural Network for image classification. Feature learning is performed by the convolutional (subsampling) and pooling layers, while the classification is finally left to a fully connected MLP.

Generative Adversarial Networks (GANs)

The core idea of GANs is to have a supervised setup with two deep NNs, one that is a generative model (the simulator) and a discriminative model, that tries to detect whether the other network is generating examples that are similar to those of the training set (Fig. 4.10). For an application to LHC physics, an alternative to more traditional simulators would be to train GANs with events from the output of a detector, while the generator can be used to sample events A recent work [47] demonstrates that it is possible to recover many physical observable distributions from dijet events by training a GAN as an alternative to the more traditional chain of simulators.

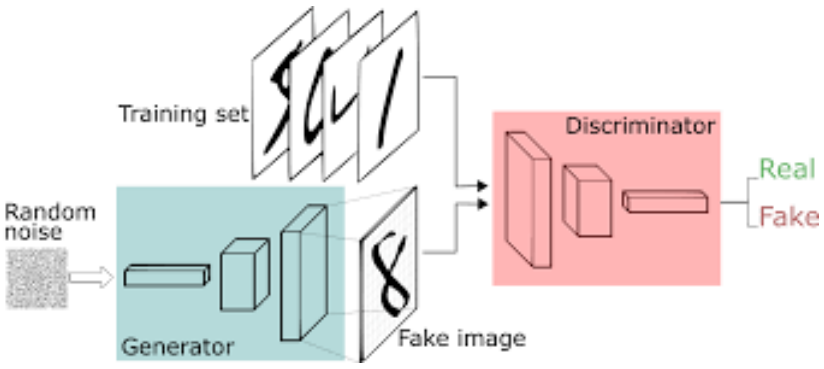

Figure 4.10: Illustration of how the generator and the discriminator work together in a GAN.

Chapter 5

Learning Algorithms

As mentioned in Section 4.1.1, the learning process consists in finding the vector of parameters θ that minimizes or maximizes a loss function $\mathcal{L}$, depending on its definition. In the following, only minimizing task is treated, being the maximizing task conceptually the same. Optimality is achieved if the global minimum, instead of a local minimum, is found. Although a few such problems can be solved exactly, optimization problems are not in general analytically solvable, but optimization methods, known as *optimizers*, often offer approximate solutions. In most cases, it is therefore acceptable to converge to a stable local minimum, instead of the global minimum. As for the loss function, also the choice of the optimizers is free. These methods can be categorized according to the level of information they use and the complexity of the calculation. Zero-order methods only use the value of the function $\mathcal{L}(\theta)$, first-order methods use the first derivative $\mathcal{L}'(\theta)$ and second-order methods also require the value of the Hessian $\mathcal{L}''(\theta)$. A class of widely used first-order methods is that of gradient-based optimizers.

5.1 Gradient-Based Optimization

Given a differentiable loss function $\mathcal{L}$, parameterized by a set of weights and biases θ, the parameters optimization can be based on the *gradient descent* (GD). It is a known result that the gradient of a function, calculated in a point, indicates the direction of fastest local ascent. In this approach, the optimal set θ is therefore searched following the direction opposite to that indicated by the gradient of the loss function. For small variations of θ, the variation of $\mathcal{L}$ can be expressed as:

$$\Delta\mathcal{L}(\theta) \simeq \frac{\partial\mathcal{L}}{\partial\theta_1}\Delta\theta_1 + ... + \frac{\partial\mathcal{L}}{\partial\theta_n}\Delta\theta_n \tag{5.1}$$

If you define the gradient of $\mathcal{L}$ as

$$\nabla_\theta\mathcal{L} = (\frac{\partial\mathcal{L}}{\partial\theta_1}, ..., \frac{\partial\mathcal{L}}{\partial\theta_n}) \tag{5.2}$$

and all the single variations of θ as

$$\Delta\theta = (\Delta\theta_1, ..., \Delta\theta_n) \tag{5.3}$$

you can synthesize the expression 5.1 as follows:

$$\Delta\mathcal{L} \simeq \nabla_\theta \mathcal{L} \cdot \Delta\theta \tag{5.4}$$

Since the loss function has to decrease, so that the accuracy of the predictions grows, the quantity $\Delta\theta$ must be negative. To do so, it is convenient do define $\Delta\theta$ as:

$$\Delta\theta = -\eta \ \nabla_\theta \mathcal{L} \tag{5.5}$$

where η is a positive parameter called *learning rate* that makes the variation of $\mathcal{L}$ negative:

$$\Delta\mathcal{L} = \nabla_\theta \mathcal{L} \cdot (-\eta \ \nabla_\theta \mathcal{L}) = -\eta \ ||\nabla_\theta \mathcal{L}||^2 < 0 \tag{5.6}$$

Moreover, it can be proved that the choice of $\Delta\theta$ in Eq. 5.5 makes $|\Delta\mathcal{L}|$ maximum, speeding up the decreasing of $\mathcal{L}$. The solution for the optimization problem is found iteratively, proceeding by steps. The set of parameters, on which $\mathcal{L}$ depends, is first initialized to a particular random point θ_0 and updated according to the rule in Eq. 5.7, which is repeated until convergence over a certain number of iterations t.

$$\theta_{t+1} = \theta_t - \lambda_t \nabla_\theta \mathcal{L}(\theta_t) \tag{5.7}$$

In this equation, each θ_t is the particular set of parameters calculated in the $t-th$ step. The learning rate is also labeled with this index because in general it is not supposed to be a constant. It is instead often convenient to define a variable learning rate, tuned according to the actual step of iteration or to the shape of the loss function. It is not necessary to use all the examples (~ 1 million) for each single iteration step. In fact, the weights are often updated at each step by using one or a small number of solved examples (~ 50), as discussed in Section 5.2. When all the examples have been used, it is said that an *epoch* has passed. The training is performed over several epochs (~ 10 or 100), by using several times the whole training dataset.

5.1.1 Learning Rate

The learning rate is a crucial problem-dependent parameter that determines how large is the step in the gradient descent. Its magnitude can be held constant during the optimization procedure, or adjusted at each iteration using various methods. If held constant, two limit cases must be considered. If it is too large it may prevent convergence to a minimum, while if it is too small, too many iterations may be needed until convergence or, in some cases, it might not be possible to escape local minima (see Fig. 5.1). Instead of searching for an empirical compromise between these two limit cases, a better strategy is to find an adaptive learning rate. A simple approach consists to adjust this parameter over time, i.e. over the epochs (t), by introducing a decay factor, such as $\eta_{t+1} = \frac{\eta_t}{t+1}$, or any other power of $t+1$. Other popular options include finite step decay, in which the decay rate is decreased by a constant factor

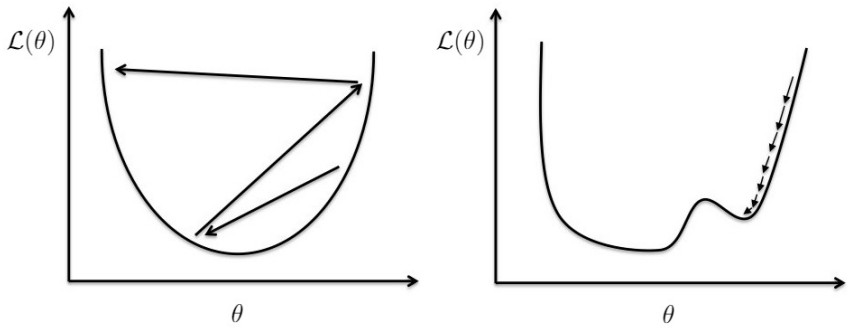

Figure 5.1: On the left: a too large value of learning rate does not allow to converge, causing overshooting. On the right: if a too small learning rate is used, a local minimum, instead of the global minimum, is reached after many iterations.

every certain number of epochs, and exponential annealing. Another option is to find a data driven η, based on the computation of the eigenvalues of the Hessian, that adapts to the loss landscape. In the last years there have been many published papers about the optimization theory for machine learning. Examples of adaptive learning rate algorithms are Adam [48], Adadelta [49], Adagrad [50], RMSProp [51] and others. Two popular algorithms are here briefly described.

RMSProp

The core idea in RMSProp (Root Mean Square Propagation) is to divide the learning rate for a weight by a running average of the magnitudes of recent gradients for that weight. So, first the running average is calculated in terms of means square:

$$v(\theta, t+1) := \gamma(\theta, t) + (1-\gamma)(\nabla \mathcal{L}_i(\theta))^2 \tag{5.8}$$

where γ is the forgetting factor, t indexes the current training iteration and i labels the observation in the training dataset. The parameters are updated as:

$$\theta := \theta - \frac{\eta}{\sqrt{v(\theta, t)}} \nabla \mathcal{L}_i(\theta) \tag{5.9}$$

RMSProp has shown good adaptation of learning rate in different applications and in particular is capable to work with mini-batches as well as with full-batches.

Adam

Adam (short for Adaptive Moment Estimation) is an update to the RMSProp optimizer. In this optimization algorithm, running averages of both the gradients (m) and the second moments (v) of the gradients are used. Correct estimators $\hat{m}$ and $\hat{v}$ are used, because m and v are biased towards values close to zero. Adam's parameter update is given by:

$$m_\theta^{(t+1)} = \beta_1 m_\theta^{(t)} + (1-\beta_1)\nabla_\theta \mathcal{L}^{(t)} \tag{5.10}$$

$$v_\theta^{(t+1)} = \beta_2 v_\theta^{(t)} + (1-\beta_2)(\nabla_\theta \mathcal{L}^{(t)})^2 \tag{5.11}$$

$$\hat{m}_\theta = \frac{m_\theta^{(t+1)}}{1-\beta_1^{t+1}} \tag{5.12}$$

$$\hat{v}_\theta = \frac{v_\theta^{(t+1)}}{1-\beta_2^{t+1}} \tag{5.13}$$

$$\theta^{(t+1)} = \theta^{(t)} - \eta \frac{\hat{m}_\theta}{\sqrt{\hat{v}_\theta + \epsilon}} \tag{5.14}$$

where ϵ is a small scalar (e.g. 10^{-8}) used to prevent division by 0, and β_1 (e.g. 0.9) and β_2 (e.g. 0.999) are the forgetting factors for gradients and second moment of gradients, respectively.

5.2 Stocastic Gradient Descent

When utilizing gradient descent to learn the optimal configuration of model parameters, the updating rule in Eq. 5.7 requires the calculation of the partial derivatives of $\mathcal{L}$ with respect to all the parameters of the model, that can be very large in number. In every single update of the model, the computer needs to keep saved all parameters values runtime; thus the memory of computer used to run the algorithm sets a limit to the complexity of the net to be processed, according to the hardware capabilities. Moreover, The loss function should be calculated over the entire training dataset. If we take, for example, typical loss functions such as mean absolute error or mean squared error, $\mathcal{L}$ has the form of a sum:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}_{x_i} \tag{5.15}$$

where $\mathcal{L}_{x_i}$ is, in these cases, the absolute or squared error calculated over one of the N training examples x_i. The gradient of $\mathcal{L}$ can be expressed as:

$$\nabla\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} \nabla\mathcal{L}_{x_i} \tag{5.16}$$

This means that the gradients in the parameter space are first calculated over all single examples x_i and then combined together in the definition of the chosen loss functions. If the number N is of the order of million events it is easy to understand that, from a computational point of view, this approach would be very expensive and would require large memories. To overcome this issue, *Stochastic Gradient Descent* (SGD) has been developed. It is a variation of the gradient descent algorithm that calculates the error and updates the model for each example in the training dataset. The stochastic mechanism takes place in each iteration by randomly pick a single example in the training dataset to get a rough estimation of the gradient. In the next iteration, a different point is extracted. This results in a feasible and fast, but also very noisy gradient approximation with high variance updates. Conversely, computing the gradient using the entirety of the training dataset yields a very reliable

estimate of the gradient, but is computationally expensive and data inefficient. A better strategy lies somewhere in between.

Mini-batch SGD

In this variant of gradient descent, instead of taking a single data point and instead of taking all of the points, the training set is randomly split into small batches that are used to calculate model error and update model coefficients. This compromise between the two extrema, allows to compute a fast and yet robust update. Equation 5.7 should be modified as follows:

$$\theta_{t+1} = \theta_t - \lambda_t \frac{1}{|B|} \sum_{x \in B} \nabla_\theta f(x, \theta_t) \tag{5.17}$$

where B identifies the set of randomly selected examples in a mini-batch, $|B|$ is its size, and x is one such example. The batch is typically made by a number between 10 and 100 points, depending on how rough or accurate the gradient should be approximated and depending on what is the desired speed for the calculation. Mini-batch SGD computes an unbiased, variable estimate of the full gradient, while reducing the variance compared to the single-sample approximation [52]. The true gradient is obtained by averaging the gradient from each of those points. It provides tangible computational advantages, because it allows to massively paralyze this computation by splitting batches on multiple GPUs or multiple computers, to achieve even more significant speed-ups.

5.3 Backpropagation

For any learning algorithm based on gradient descent, the loss function must be differentiable with respect to its parameters. The analytical computation of the gradients is a complex task, especially in deep learning. The *backpropagation* algorithm is a widely used solution in training feed-forward neural networks for supervised learning purpose. This method computes in an efficient way the effect that each intermediate neuron, from the input to the output layer, had in the final loss. The chain rule is applied to compute the gradient with respect to each weight for a fixed input–output pair (x_i, y_i). However, doing this separately for each weight would be inefficient. Backpropagation computes the gradient one layer at a time, iterating backward from the last layer to avoid redundant calculations of intermediate terms in the chain rule, maximizing sub-expression reuse [53]. For the purpose of backpropagation, the specific loss function and activation functions do not matter, as long as they and their derivatives can be evaluated efficiently.

5.3.1 Matrix Multiplication

In the case of a feed-forward network with a scalar loss function, backpropagation can be simply treated by means of matrix multiplication [54]. Backpropagation evaluates the expression for the derivative of the loss function as a

product of derivatives between each layer from left to right, with the gradient of the weights between each layer being a simple modification of the partial products (the "backwards propagated error"). Once defined the weighted input of each layer as z^l and the output of layer l as the activation a^l, the derivative of the loss in terms of the inputs is given by the chain rule:

$$\frac{d\mathcal{L}}{da^L} \cdot \frac{da^L}{dz^L} \cdot \frac{dz^L}{da^{L-1}} \cdot \frac{da^{L-1}}{dz^{L-1}} \cdot \frac{dz^{L-1}}{da^{L-2}} \cdots \frac{da^1}{dz^1} \cdot \frac{\partial z^1}{\partial x} \tag{5.18}$$

These terms are the derivative of the loss function, the derivatives of the activation functions and the matrices of weights.

$$\frac{d\mathcal{L}}{da^L} \cdot (g^L)' \cdot W^L \cdot (g^{L-1})' \cdot W^{L-1} \cdots (g^1)' \cdot W^1 \tag{5.19}$$

This expression can be simplified by a suitable choice of activation functions, whose derivative should be as simple to calculate as possible, as mentioned in Sec. 4.2.1. The gradient ∇ is the transpose of the derivative of the output in terms of the input, so the matrices are transposed and the order of multiplication is reversed, but the entries are the same:

$$\nabla_x \mathcal{L} = (W^1)^T \cdot (g^1)' \cdot \cdots \cdot (W^{L-1})^T \cdot (g^{L-1})' \cdot (W^L)^T \cdot (g^L)' \cdot \nabla_{a^L} \mathcal{L} \tag{5.20}$$

Backpropagation consists essentially in the evaluation of this expression from right to left, computing the gradient at each layer on the way. An auxiliary quantity δ^l for the partial products is introduced, interpreted as the error at level l and defined as the gradient of the input values at level l:

$$\delta^l := (g^l)' \cdot (W^{l+1})^T \cdot \cdots \cdot (W^{L-1})^T \cdot (g^{L-1})' \cdot (W^L)^T \cdot (g^L)' \cdot \nabla_{a^L} \mathcal{L} \tag{5.21}$$

The quantity δ^l is a vector of length equal to the number of nodes in level l. The gradient of the loss with respect to the weights in layer l is then:

$$\nabla_{W^l} \mathcal{L} = \delta^l (a^{l-1})^T \tag{5.22}$$

The factor of a^{l-1} is because the weights W^l between level $l-1$ and l affect level l proportionally to the activations in input. The δ^l can easily be computed recursively from the previous layer:

$$\delta^{l-1} := (g^{l-1})' \cdot (W^l)^T \cdot \delta^l \tag{5.23}$$

The gradients in the weights space can thus be computed using a few matrix multiplications for each level. Backpropagation avoids inefficiency in two ways. Firstly, it avoids duplication because when computing the gradient at layer l, it is not needed to recompute all the derivatives on later layers $l+1, l+2, ...$ each time. Secondly, it avoids unnecessary intermediate calculations because, at each stage, it directly computes the gradient of the weights with respect to the ultimate output (the loss), rather than unnecessarily computing the derivatives of the values of hidden layers with respect to changes in weights $\partial a^{l'}_{j'} / \partial w^l_{jk}$.

Chapter 6

Data Analysis

In this chapter, the concepts illustrated so far are applied to the data analysis. A regression DNN is built to estimate the invariant mass m_{tt} of the top-antitop quark pairs decaying in the dilepton channel, having as inputs the measurements of the decay products. Such an estimate, performed event-by-event, could be important for different measurements (like differential $t\bar{t}$ measurements) and searches for BSM physics with $t\bar{t}$ final states. In particular, as described in Section 2.3, $t\bar{t}$ resonances appear in different BSM scenarios, and to identify them, the best possible $m_{t\bar{t}}$ estimate is essential, in order to resolve the distinctive peak in the observed spectrum. As commented already in Section 2.2.3, the full $t\bar{t}$ system reconstruction by relying only on the visible measured final-state particles is particularly challenging in the dilepton channel, due to the presence of two undetected neutrinos that contribute together to the missing transverse energy.
All the datasets used in this chapter, for both training and testing, are Monte Carlo (MC) samples that simulate the $t\bar{t}$ pair production emerging from p-p collisions (as it happens at the LHC), the further decay in the dilepton channel and the detection of the final states by the ATLAS detector. These MC samples are produced by the ATLAS collaboration and undergo a typical events selection as that applied to real data events used to study dilepton $t\bar{t}$ events and to search for $t\bar{t}$ resonances in this channel.
This chapter is organized as follows: the hardware and software setups, used throughout the data analysis, are described in Section 6.1. The MC sample used to train the DNN is analyzed in Section 6.2. From this sample, two different training datasets are extracted. In the first one, the m_{tt} follows the physical SM distribution provided by the simulation. The second set is obtained applying data augmentation and re-sampling, in order to have a more uniform m_{tt} distribution. The architecture of the model that showed the best performance is illustrated in Section 6.3. The criteria that led to this choice and a comment on the other models that have been tested are also given. The training phase is then discussed in Section 6.4. This procedure was repeated twice, with the same optimal model architecture and set of hyper-parameters, changing only the training sets. Once the training is completed, the application phase for both the two models takes place (Section 6.5) on two different simu-

lated testing sets: in the first one the mass follows the SM distribution, while the second set represents events coming from the production and decay of a hypothetical heavy $t\bar{t}$ resonance. In both testing cases, the model can use only partial information to perform the regression, given by the detected particles, due to the presence of the two undetected neutrinos. Moreover, these complex final states include hadronic jets, which are difficult to distinguish from other collision event products. For these reasons the DNN is not expected to solve the problem exactly. A comparison with respect to more traditional methods is also given.

6.1 Hardware and Software Setups

The machine on which the programs were run was provided by the INFN[1] farm for scientific computing of Trieste, based on Linux systems. No Graphic Processing Unit (GPU) were available, and this sets a limit to DL applications. The analysis here discussed is performed by means of Python-based programs. Python [55] is an interpreted high-level programming language that allows the usage of several libraries written to meet the needs of almost all the scientific areas. In particular I used Pandas [56] to handle datasets, NumPy [57] to perform mathematical calculations and Matplotlib [58] to plot the graphs. Since the last few years saw a rapid increase in interest in artificial intelligence, there was a subsequent releases of new open source libraries, which provide a flexible interface for neural networks modeling and training. In this thesis I chose to use Keras [59] together with TensorFlow [60] (CPU-only version) as backend. Keras is a high-level neural networks Application Programming Interface (API), written in Python. This library contains all the basic computational tool necessary to build and run a ML algorithm based on neural networks. It allows to choose between different kind of models, based on feed-forward, convolutional, recurrent layers and many others. The principal learning methods have already been implemented too. Keras is capable of running, among the other underlying frameworks, on top of TensorFlow, which is a symbolic math library based on dataflow and differentiable programming, that manages the tensor calculations.

6.2 Training Datasets

The datasets used for the DNN training and testing consist of events generated through MC simulation, including the full simulation of the ATLAS detector, performed with Geant4 [61]. The dataset used for training represents $t\bar{t}$ events with dilepton decays, from p-p collisions at $\sqrt{s} = 13$ TeV, with the LHC Run 2 beam conditions. These events are generated at the NLO-QCD accuracy by the Powheg MC program [62], interfaced to Pythia8 for simulating the parton-shower and hadronization steps [63]. A basic event selection is then applied to the simulated events, requiring the presence of at least 2 reconstructed hadronic

[1]The italian National Institute for Nuclear Physics.

jets and exactly two high-transvere-momentum opposite-charged leptons (electrons or muons). After the event selection, the dataset consists of about 40 million events, grouped in three separate sets, according to whether the lepton pair is given by two e (20.12%), two μ (30.18%) or one e and one μ (49.70%). These percentages differ with respect to the theoretical branching ratios, because of the different selection efficiencies for events in the three subchannels. Each row in the training set consists of a reconstructed $t\bar{t}$ event and carries one label (the correct $m_{t\bar{t}}$) and a fixed amount of 22 features, which correspond to the measured four momenta of the two charged leptons and of up to three jets in the event, as well as the available information on the missing energy in the transverse plane. The four momenta are expressed in terms of four variables widely used in experimental particle physics, well calibrated and reconstructed from the energy deposits in the various detectors, which are the energy (E), the transverse momentum (p_T), the azimuthal angle (ϕ) and the pseudorapidity (η), measured for each of the two leptons and three jets, in addition to the missing transverse energy (missing E_T) and the associated angle (missing ϕ). Being the number of columns fixed, if less than three jets are produced, the missing measurements are filled with the 0 value. A schematic representation of how the DNN, given input data, produces the output, is shown in Fig. 6.1. The usage of such low level features is justified by the choice of a DNN

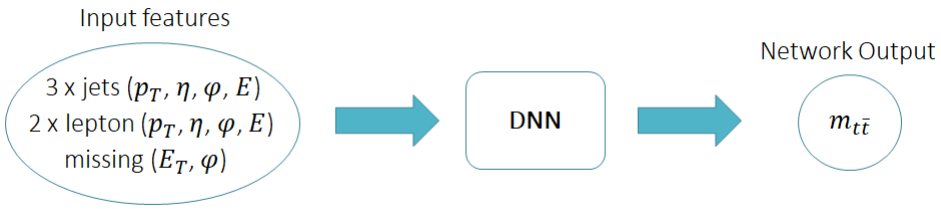

Figure 6.1: For each event, the DNN receives as input 22 features, which are the quantities measured by the detector, and performs a regression to estimate the $m_{t\bar{t}}$.

architecture. Deep learning has in fact the property to automatically extract the useful pieces of information from the dataset, without requiring any preprocessing. This means that there is no need to combine such features, even if other kind of high level features could be useful to perform the regression, such as *b-tagging*, for example. This has not been implemented in this thesis, but it may be in future studies. From this MC sample, two datasets were extracted, respecting the proportions between the three subchannels.

"Peak" Training Data

The first dataset is simply obtained by extracting 6 million events from the initial sample. The histogram of the target is plotted in Fig. 6.2 and follows the distribution predicted by the SM. There is a peak at about 500 GeV, around which most of the measurements are distributed, which is followed by an exponential decay. Consequently, there are few events with mass greater than 1 TeV, which is one of the regions where one expects to observe the

interesting events of BSM physics. For this reason, even if a neural network trained on this dataset is able to correctly describe similarly distributed test data, it is not certain that it would be able to reconstruct correctly the sought BSM events as well. The DNN is fact trained to "prefer" $m_{t\bar{t}}$ values around 500 GeV. This consideration motivated the idea to build also a second set, which has a more uniform m_{tt} distribution in a wider range. The histograms of the features are instead plotted in Fig. A.1 in Appendix A. In most events, only two jets are produced and the missing values, corresponding to the third jet, are filled with the 0 value.

"Uniform" Training Data

This second dataset was obtained by sampling the MC simulation in bins with a width of 100 GeV. The code used to do this sets a maximum number of events that can be accepted in each bin, which is filled accordingly. Starting with about 40 million events, this procedure selects a maximum amount of about 3 million events. By exploiting the symmetry of the physical problem, it was then possible to double the number of events by simultaneously inverting all the ϕ angles. This type of procedure is usually known as *data augmentation* and is widely used in machine learning. In the previous dataset, the same number of events was selected retrospectively, in order to study the effect due only to the quality of the data, and not to the quantity. In both cases, the effect of the training dataset size on loss optimization is discussed Section 6.4.1, to understand if an optimal amount of examples has been chosen or if it has been instead underestimated or overestimated. The resulting $m_{t\bar{t}}$ distribution is plotted in Fig. 6.2. It is possible to see several peaks, which are actually

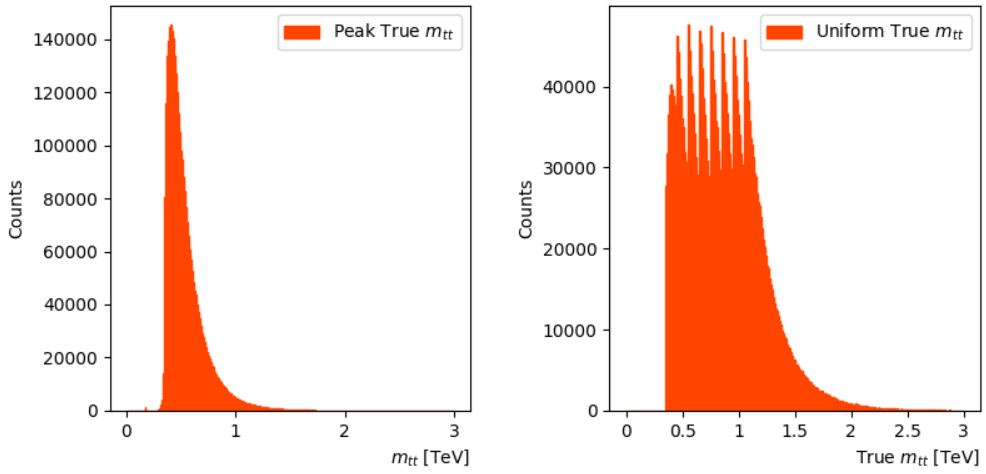

Figure 6.2: On the left: histogram of the true $m_{t\bar{t}}$ for the "peak" dataset. On the right: same histogram for the "uniform" dataset. Both samples share the same amount of events, which is 6 million.

decreasing exponentials, because more bins were used in this histogram than in the data sampling, where the bins were large enough to show an exponential distribution of the data. The histogram is approximately uniform up to 1.2

Layer	Connection	Output shape	Parameters
Hidden (1)	Dense	(None, 100)	2300
Hidden (2)	Dense	(None, 100)	10100
Hidden (3)	Dense	(None, 100)	10100
Output	Dense	(None, 1)	101

Table 6.1: Schematic summary of the model chosen for data analysis.

TeV, which is a value chosen to be greater than the invariant mass of BSM test events analyzed in Section 6.5. Then, also in this case an exponential decay follows, with a non-negligible amount of events up to about 1.5 TeV. In Fig. A.2 in Appendix A, the distributions of the features are shown. There are no artifacts that significantly change the shape of the distributions, except clearly for the energy and transverse moment tails, which in this case are higher because they are correlated with a higher m_{tt}.

6.3 Model Architecture

The model used in the following analysis consists of a deep feed-forward neural network. This means that all nodes in adjacent hidden layers are fully connected (dense layers). The number of layers and the amount of neurons in each layer were chosen empirically, after many attempts. The predictive power of a neural network initially tends to increase with the network complexity, which can be roughly estimated by the number of parameters. However, models that are too large fail to generalize, as they use their parameters to memorize the input-output pairs given in the solved examples, rather than to find a general rule capable of solving the task. This issue is generally known as *overfitting* and is determined not only from the architecture of the model, but also by the amount and quality of solved examples in the training set. Furthermore, performance usually vary gradually together with the model complexity, consequently one usually reaches a plateau for the predictive power of the network, before starting to overfit. For this reason, as a general rule, once one has found a set of models that provide similar performance, it is a good practice to avoid the use of models that are too large, that is, which have many parameters, preferring simpler architectures. A summary of the simplest and at the same time more performing model is shown in Table 6.1. The model is made up by three hidden layers, each one with 100 neurons. The total number of parameters, which are all trainable, is 22601. Many other models were tested changing only the network structure, therefore keeping unchanged the training set and all parameters. Up to 10 hidden layers and up to 200 hidden neurons were tested. Configurations with different numbers of neurons in each layer were also tested. I saw in general that models with less than 100 neurons per layer achieve lower performance with respect to the chosen model, i.e. the loss reached a higher asymptotic value, while models with more than 100 neurons

per layer tend to increase the overfitting, which becomes evident using 200 nodes. Maintaining 100 neurons per layer and increasing the number of layers up to 5, I got indistinguishable performance. Moreover, very deep networks (from 5 to 10 hidden layers) have similar performance but tend also to increase the overfitting, due to the poor training sample statistics compared to the network complexity.

6.4 Training Phase

Before to train the DNN, a set of hyper-parameters must be chosen. Such parameters heavily affect the training, as already described in the previous chapters. Their values, which were found empirically, are here listed.

- **Activation function**: ReLU
- **Loss**: Mean Absolute Error (MAE)
- **Epochs**: 50
- **Optimizer**: Adam
- **Learning rate**: 0.0001
- **Mini-batch size**: 50

For input and hidden layers a ReLU activation function, which is the most popular option nowadays, has been set, while for the output a linear activation function was chosen. Also Exponential Linear Unit (ELU) as activation function and Mean Squared Error (MSE) as loss were tested, without any improvements. During the loss optimization, the training set is randomly split in two groups for training and validation purpose. 80% of the training set was used to actually train the network, i.e. the loss is minimized on these data (*train* curve in Fig. 6.3 and 6.4). The remaining 20% of the data was used only to evaluate the loss (*val* curve), which was not optimized on this set. This allows to estimate the presence and the amount of overfitting, which occurs when loss and validation loss start to diverge. In case of overfitting, the loss will continue to decrease, while the validation loss will not follow the same trend. Considered that different choices for the loss function are possible and that each one can give different results, an alternative metrics is usually also considered to evaluate the models. This second function was calculated on both training and validation data in each epoch, without being optimized. In the following, the MSE was chosen as metrics. The training performed using the "peak" dataset (*peak DNN* in the following) is shown in Fig. 6.3, while in Fig. 6.4 the "uniform" dataset (*uniform DNN*) was instead used. In both cases the overfitting, which is in general always present in some amount, is very small. Validation curves are noisier than training curves due to the lower amount of data used to calculate them. Both loss functions reach similar asymptotic values, while the main difference is in the metrics, which is lower

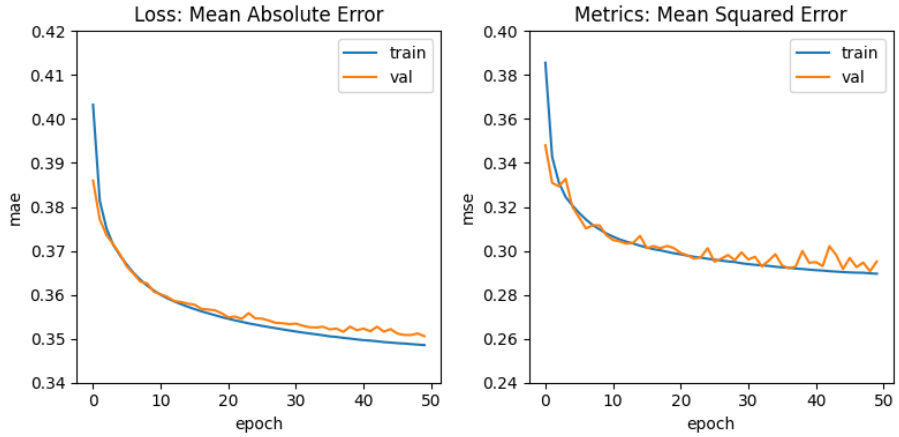

Figure 6.3: Training performed using the "peak" dataset. The curve on the left represents the loss function, while on the right the alternative metrics is evaluated.

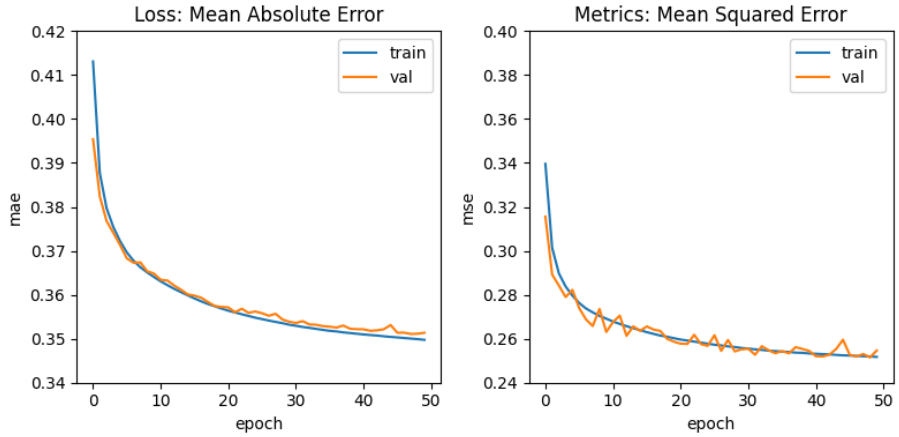

Figure 6.4: Training performed using the "uniform" dataset. The curve on the left represents the loss function, while on the right the alternative metrics is evaluated.

in the uniform model. The number of epochs is also chosen empirically and set to 50, which is the number of times the DNN uses the entire training dataset. This particular value can be considered as the beginning of the training and validation loss plateaus, which persist for several hundred epochs. Adam is chosen as optimizer. This adaptive method requires to declare an initial value, which is set to 0.0001. This value might seem very small, but larger values result in unstable loss. Finally, the mini-batch size does not seem to heavily affect the training, so it is set to its default value (50).

6.4.1 Training Dataset Size

The amount of solved examples is also an important parameter in ML applications. An adequate dataset size can help to reduce both the minimum loss and the overfitting. In general, there are no drawbacks in using an excessively large dataset, except that the time required to perform the training increases

dramatically with the dataset size. To understand if an optimal value was chosen, the two training ("peak" in Fig. 6.5 and "uniform" in Fig 6.6) were repeated over datasets with different size. For simplicity only the loss curves are shown. In both cases, the training was significantly improved going from

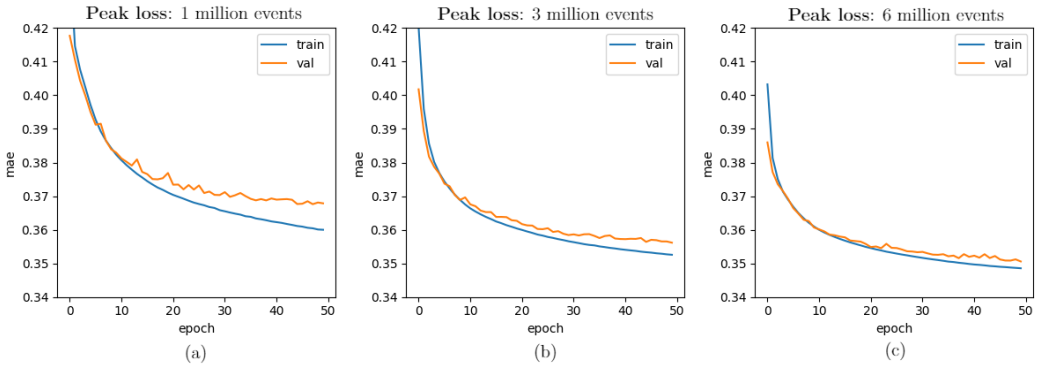

Figure 6.5: "Peak" loss curves plotted with increasing dataset size: 1 million (a), 3 million (b) and 6 million (c) events.

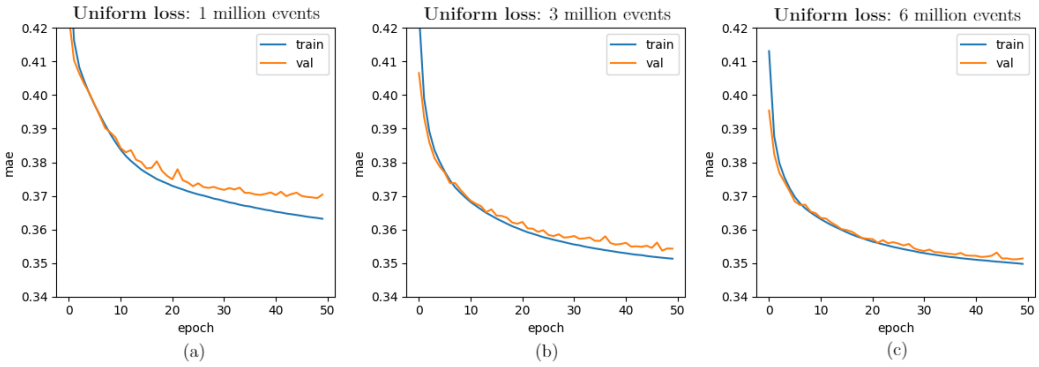

Figure 6.6: "Uniform" loss curves plotted with increasing dataset size: 1 million (a), 3 million (b) and 6 million (c) events.

1 million to 3 million data, while the effect is less important by doubling the dataset size to 6 million events. From this study it can be concluded that further increasing the training dataset size would not be an effective strategy to significantly improve the performance of the DNNs.

6.5 Testing Phase

The DNNs are now applied for $m_{t\bar{t}}$ reconstruction in two different scenarios, both simulated. The first testing set takes into account SM $t\bar{t}$ events (analogous to the events using in the training pahse), while in the second set a BSM mechanism was considered: the production of a $t\bar{t}$ pair originating from the hypothetical Z' decay. The results given by the two DNNs are compared with those given by two other more traditional methods, which are briefly described below.

llbb

This method consists in the calculation of the partial invariant mass m_{llbb} of the top-antitop pair, given by:

$$m_{llbb} = \sqrt{(p^{l^+} + p^{l^-} + p^{b_1} + p^{b_2})_\mu (p^{l^+} + p^{l^-} + p^{b_1} + p^{b_2})^\mu} \tag{6.1}$$

where $p^{l^\pm}$ and $p^{b_{1,2}}$ are the four-momentum vectors for electron or muon and b-tagged jets respectively. This method could be preferred because of its simplicity with respect to more complex methods to fully reconstruct the $m_{t\bar{t}}$, for instance the widely used neutrino weighting technique [64].

Neutrino Weighting

Given the measured momenta of leptons, jets and missing transverse energy $\not{E}_T$, as well as the available constraints from the values of M_W and m_t, the neutrino four-momenta can be determined, ideally allowing for a full reconstruction of the $t\bar{t}$ system in the dilepton channel. The neutrino weighting technique relies on scanning over a set of possible values for the two neutrino pseudorapidities, solving a system of equations for each possible pair of values, and then checking the compatibility of the found solutions with the measured value of $\not{E}_T$, by assigning a *weight* to each pair of neutrino pseudorapidities. Finally, the solution with the largest weight is selected. The experimental jet energy resolution as well as the intrinsic W-boson and top-quark mass resolutions are taken into account by energy and mass smearing [65]. The neutrino weighting method is then able to reconstruct the full $t\bar{t}$ system, and therefore to assign to each event an estimated value of $m_{t\bar{t}}$, but its performance suffer from the non-negligible fraction of events where no suitable solution is found, especially for high values of $m_{t\bar{t}}$, as well as in general degraded resolution at high $m_{t\bar{t}}$.

The performance of the various methods are now compared. First of all, the distributions of the predicted values are shown and compared with the correct (true) ones. The true $m_{t\bar{t}}$ for both testing sets is shown in Fig. 6.7. In these histograms, as well as in the following ones, the y axis measures the fraction of events per bin instead of the event counts. A comparison between the $m_{t\bar{t}}$ distributions reconstructed by the two DNNs is shown in Fig. 6.8, where

the true distributions are also shown with a dashed line. By using different training datasets, the performance ere affected: the peak DNN reconstructs an excess of SM events around the peak of the true distribution. This could be due to the fact that, since most of the measurements in the training dataset are distributed around the peak, the DNN, when "in trouble", tends to prefer or "guess" values around the peak. The BSM events distribution in the peak DNN is much wider than the true distribution and significantly shifted towards lower values. On the other hand, the uniform DNN does not provide an excess of events within the peak in the SM case. BSM events look less biased and more similar to the correct ones, though they are not reconstructed exactly (the distribution is wider). The results given by the other two methods are shown in Fig. 6.9. The llbb method suffers from an obviously large bias, because the contribution of the two neutrinos is just neglected, and the BSM distribution is significantly wider than the true distribution. The neutrino weighting predictions look quite accurate in the SM data range, despite this method gives a zero output when it cannot find a compatible solution (10% of the times in this case). This happens more frequently at high invariant mass (15% of the times in the case of the BSM sample), and even if the BSM distribution peak is reconstructed correctly, the distribution shows a long tail toward lower values.

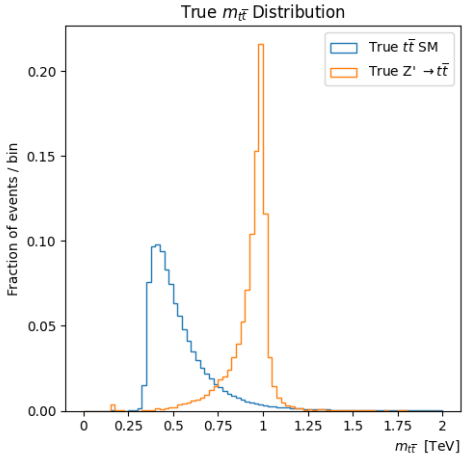

Figure 6.7: True $m_{t\bar{t}}$ distribution for SM and BSM events in the testing sets.

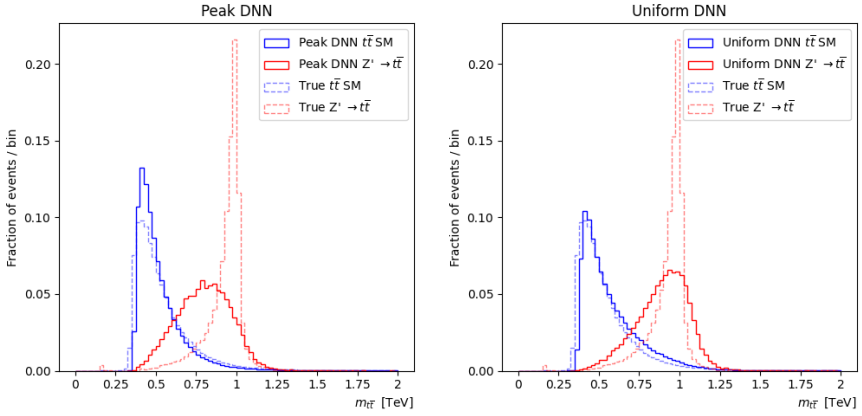

Figure 6.8: $m_{t\bar{t}}$ reconstructed by the peak (left) and uniform (right) DNN, compared with the correct distributions (dashed line).

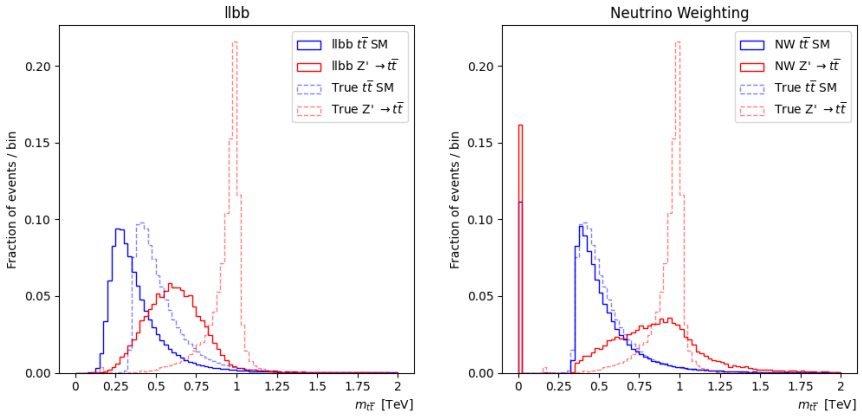

Figure 6.9: $m_{t\bar{t}}$ reconstructed by the llbb (left) and neutrino weighting (right) methods, compared with the correct distributions (dashed line).

Residuals are also calculated to more precisely estimate the accuracy of the various methods. For each method and for each testing set, they are calculated in two different ways: by taking the absolute difference between the expected and true $m_{t\bar{t}}$ in each event, and then dividing this quantity by the true value (relative difference). Their distributions are plotted in Fig. 6.10 - 6.14, which show also means and standard deviations of the residual distributions.

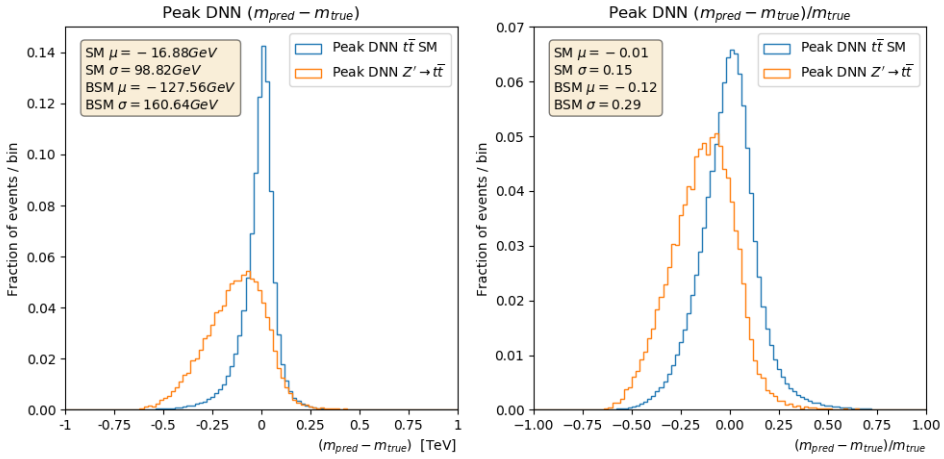

Figure 6.10: Peak DNN: residuals calculated by making the difference between the expected and true $m_{t\bar{t}}$ (left) and dividing this quantity by the true value (right).

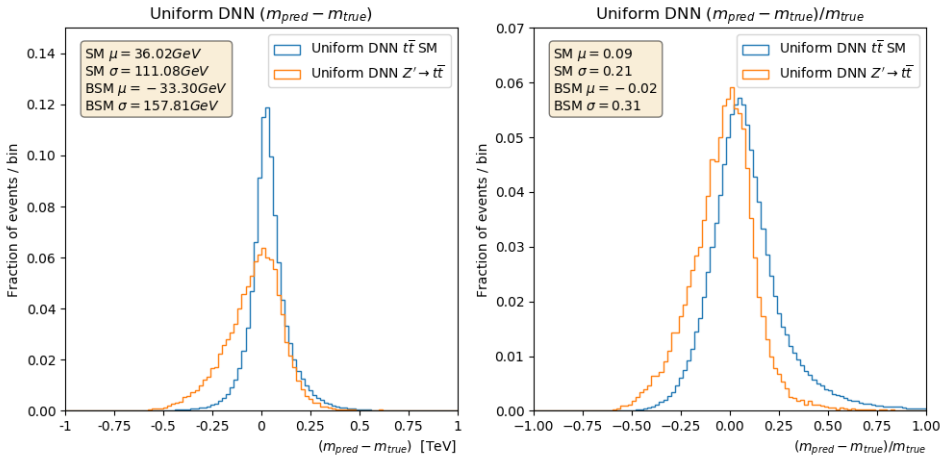

Figure 6.11: Uniform DNN: residuals calculated by taking the difference between the expected and true $m_{t\bar{t}}$ (left) and by dividing this quantity by the true value (right).

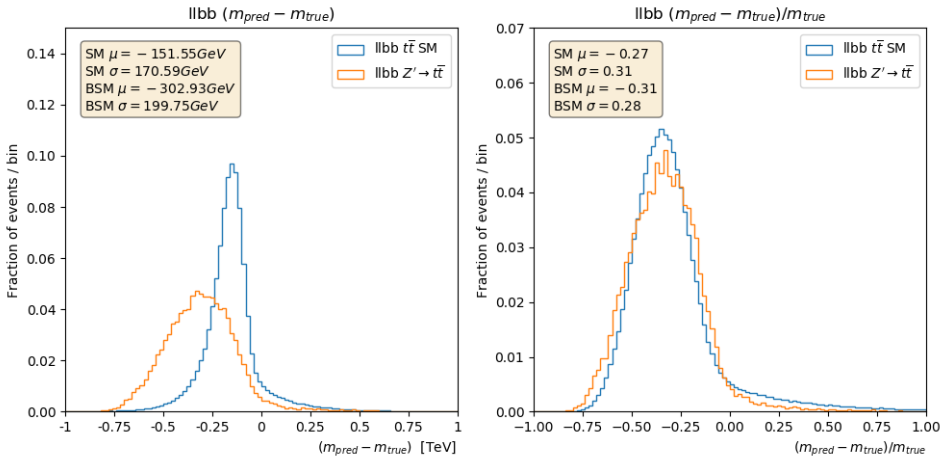

Figure 6.12:

Figure 6.13: llbb method: residuals calculated by taking the difference between the expected and true $m_{t\bar{t}}$ (left) and by dividing this quantity by the true value (right).

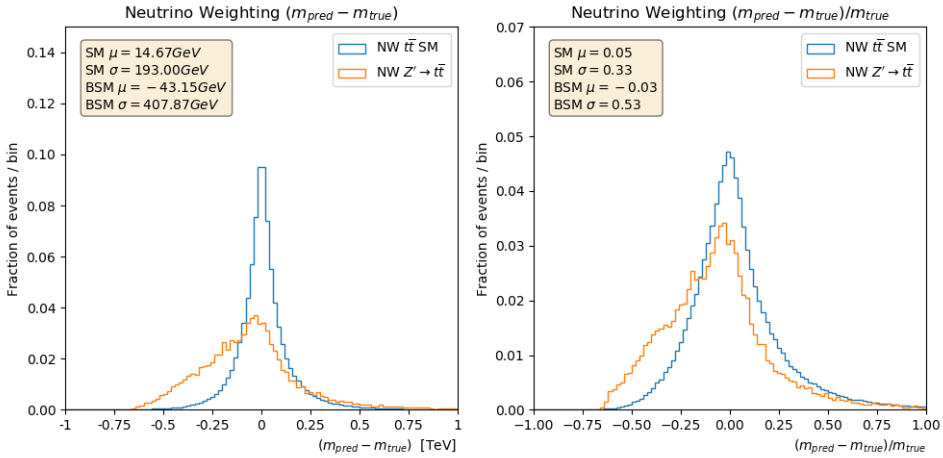

Figure 6.14: Neutrino weighting method: residuals calculated by taking the difference between the expected and true $m_{t\bar{t}}$ (left) and by dividing this quantity by the true value (right). The data corresponding to the null predictions were not considered in this plot.

In Fig. 6.14, the residuals corresponding to the null predictions are not considered, because they are invalid solutions and because this would have altered the estimates of μ and σ. However, these measurements are still included in the fraction of events calculation. In this way, the numbers shown in the y axis correspond to the actual fractions, but their sum over all bins will not be equal to one. The parameters of the residual distributions quantify more precisely what has already been said observing the distributions of the predictions. Moreover, in all cases, the Pearson correlation coefficient ρ between true and predicted values was calculated. Its definition is given in Eq. 6.2.

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} \tag{6.2}$$

	Peak DNN	**Uniform DNN**	**llbb**	**NW**
μ_{SM} (GeV)	-16.88	36.02	-151.55	14.67
σ_{SM} (GeV)	98.82	111.08	170.59	193.00
μ_{SM}^{rel}	-0.01	0.09	-0.27	0.05
σ_{SM}^{rel}	0.15	0.21	0.31	0.33
ρ^{SM}	0.85	0.82	0.59	0.61
μ_{BSM} (GeV)	-127.56	-33.30	-302.93	-43.15
σ_{BSM} (GeV)	160.64	157.81	199.75	407.87
μ_{BSM}^{rel}	-0.12	-0.02	-0.31	0.03
σ_{BSM}^{rel}	0.29	0.31	0.28	0.53
ρ^{BSM}	0.44	0.43	0.28	0.23

Table 6.2: Summary of all parameters measured in the data analysis. The four methods were evaluated on the two test datasets (SM and BSM).

where $cov(X, Y)$ is the covariance of the two variables X and Y and $\sigma_{X,Y}$ are their standard deviations. It has a value between +1 and -1, where +1 corresponds to the total positive linear correlation (the ideal value in this case), 0 means no linear correlation, and -1 is total negative linear correlation. The parameters calculated so far in the various cases are summarized in Table 6.2. The correlation coefficient is similar in both DNNs and is significantly higher than the values determined by the other two methods. This indicates a clear improvement in the $m_{t\bar{t}}$ reconstruction. On the other hand, the relatively small value of the correlation coefficient in the BSM scenario doesn't indicate bad performance: it is indeed simply due to the fact that, as already shown in Fig. 6.8, the $m_{t\bar{t}}$ distribution for the Z' signal is strongly peaked around the value of 1 TeV, with an resolution which is better than that of the DNN reconstruction. In Table 6.2, the quantities μ (GeV) and σ (GeV) refer to the absolute residuals (Fig. 6.10 - 6.14 left), while μ^{rel} and σ^{rel} refer to the relative ones (Fig. 6.10 - 6.14 right).
In Fig. 6.15 - 6.18, the 2d histograms of the two quantities are reported. The coefficient ρ and a dashed line corresponding to the ideal linear relation are also shown. It is possible to see how points are less scattered and distributed more symmetrically around the bisector, in cases where deep learning has been applied. These graphs, in addition to the previous considerations, help to indicate DNNs as a valid tool for regression task.

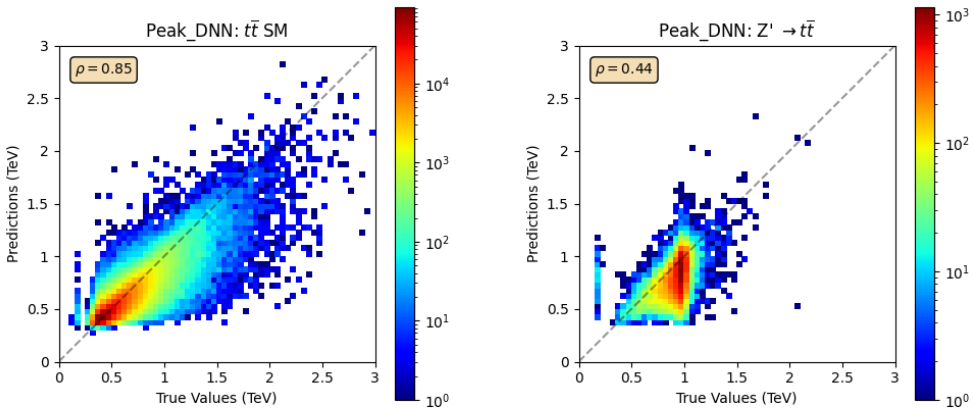

Figure 6.15: Peak DNN: 2d histogram of the true and predicted $m_{t\bar{t}}$. The Pearson correlation coefficient ρ and a dashed line corresponding to the ideal linear relation are also shown.

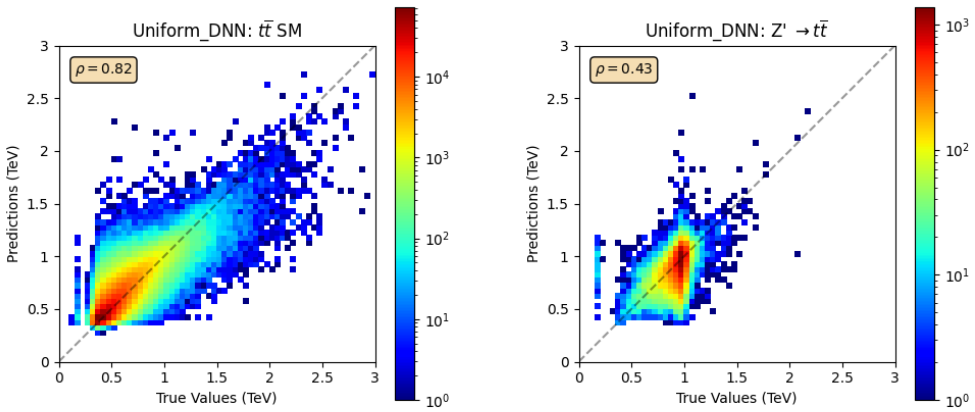

Figure 6.16: Uniform DNN: 2d histogram of the true and predicted $m_{t\bar{t}}$. The Pearson correlation coefficient ρ and a dashed line corresponding to the ideal linear relation are also shown.

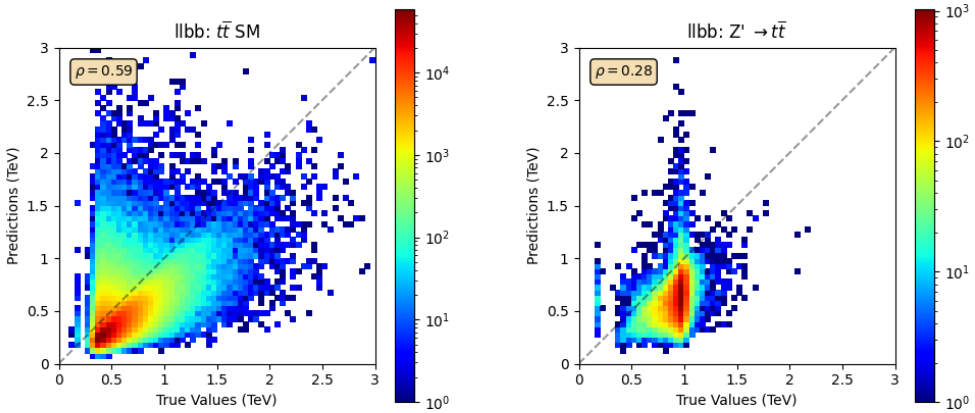

Figure 6.17: llbb: 2d histogram of the true and predicted $m_{t\bar{t}}$. The Pearson correlation coefficient ρ and a dashed line corresponding to the ideal linear relation are also shown.

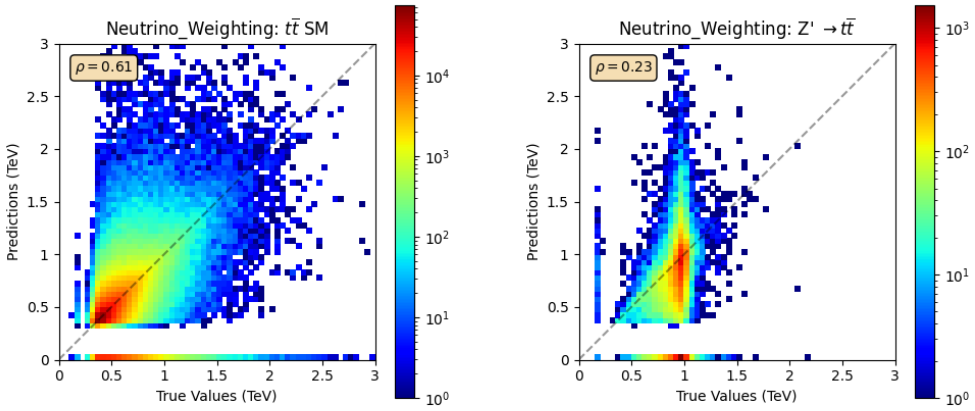

Figure 6.18: Neutrino weighting: 2d histogram of the true and predicted $m_{t\bar{t}}$. The Pearson correlation coefficient ρ and a dashed line corresponding to the ideal linear relation are also shown.

Chapter 7

Conclusions

This thesis consisted of a feasibility study, in which an innovative data analysis tool was tested in the reconstruction of the invariant mass of top quark pairs decaying in the dilepton channel. Deep learning, a recently developed technique of multivariate analysis, was applied to perform the regression in simulated ATLAS samples. In the theoretical sections all the concepts useful for understanding the final data analysis were provided. First of all, the physics framework was discussed: the SM, the top quark physics and its role in some BSM theories. Then also the LHC and the ATLAS detector, that make up the experimental setup, were illustrated. Later, the basic concepts in deep learning were given step by step. Finally, deep learning was successfully implemented in the feed-forward architecture and it proved to be a useful tool for reconstructing the invariant mass of top quark pairs decaying in the dilepton channel. Comparing this innovative method with some of the most commonly used traditional analysis techniques, such as the usage of the invariant mass of the visible decay products only ("llbb") as a proxy for $m_{t\bar{t}}$, as well as a partially analytic solution for neutrino reconstruction (the so-called "neutrino-weighting" method), it was possible to show significant improvements in mass estimation in both SM and BSM scenarios. Moreover, the idea of making a selection of the training data allowed to compare two different deep neural networks. Once applied to the BSM events of interest, given by the decay of the hypothetical Z′ boson with mass 1 TeV, there were some differences in performance. Indeed, by using the training data with a more uniformly distributed target, the mass reconstruction is less biased and more accurate, especially in the case of the high-mass range, interesting for BSM searches. However, the random errors of the two DNNs are very similar and this could mean that the possibilities for improvement are rather limited, considering that, as it is known, the problem cannot be solved exactly in this channel. Furthermore, it is believed that an optimal model and set of hyper parameters, including the amount of training examples, have been found, so there shouldn't be noticeable improvements in changing them. The starting point for future studies could be the use of new high-level features, such as b-tagging for example. The analysis could be also extended to events with more than three jets in the final states. Another strategy could be the application of a different type of neural network

than a simple feed-forward architecture. In addition to further improving the invariant mass estimate, one of the future aims will be to reconstruct simultaneously multiple targets useful for BSM physics searches. For example, in addition to the $m_{t\bar{t}}$, a variable sensitive to the spin correlation between two top quarks could be reconstructed. Moreover, the four-momentum of the two top quarks could also be reconstructed, in order to extract differential measures of different variables with a single DNN.

Appendix A

Training Features

The histograms of the 22 features that make up the "peak" and "uniform" training datasets are shown respectively in Fig. A.1 and A.2.

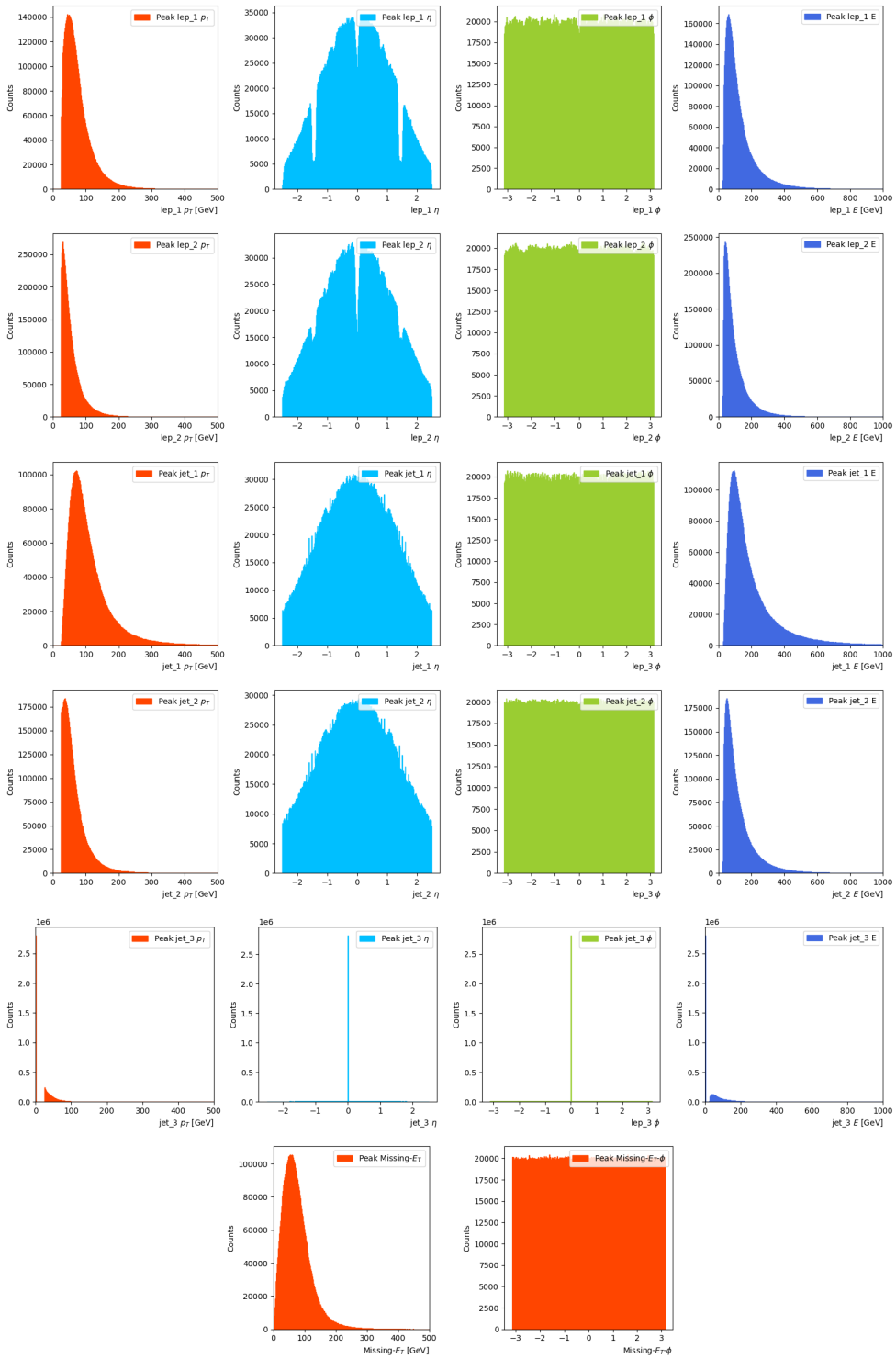

Figure A.1: Histograms of the 22 features that make up the "peak" dataset. In most events, only two jets are produced and the missing values, corresponding to the third jet, are filled with the 0 value.

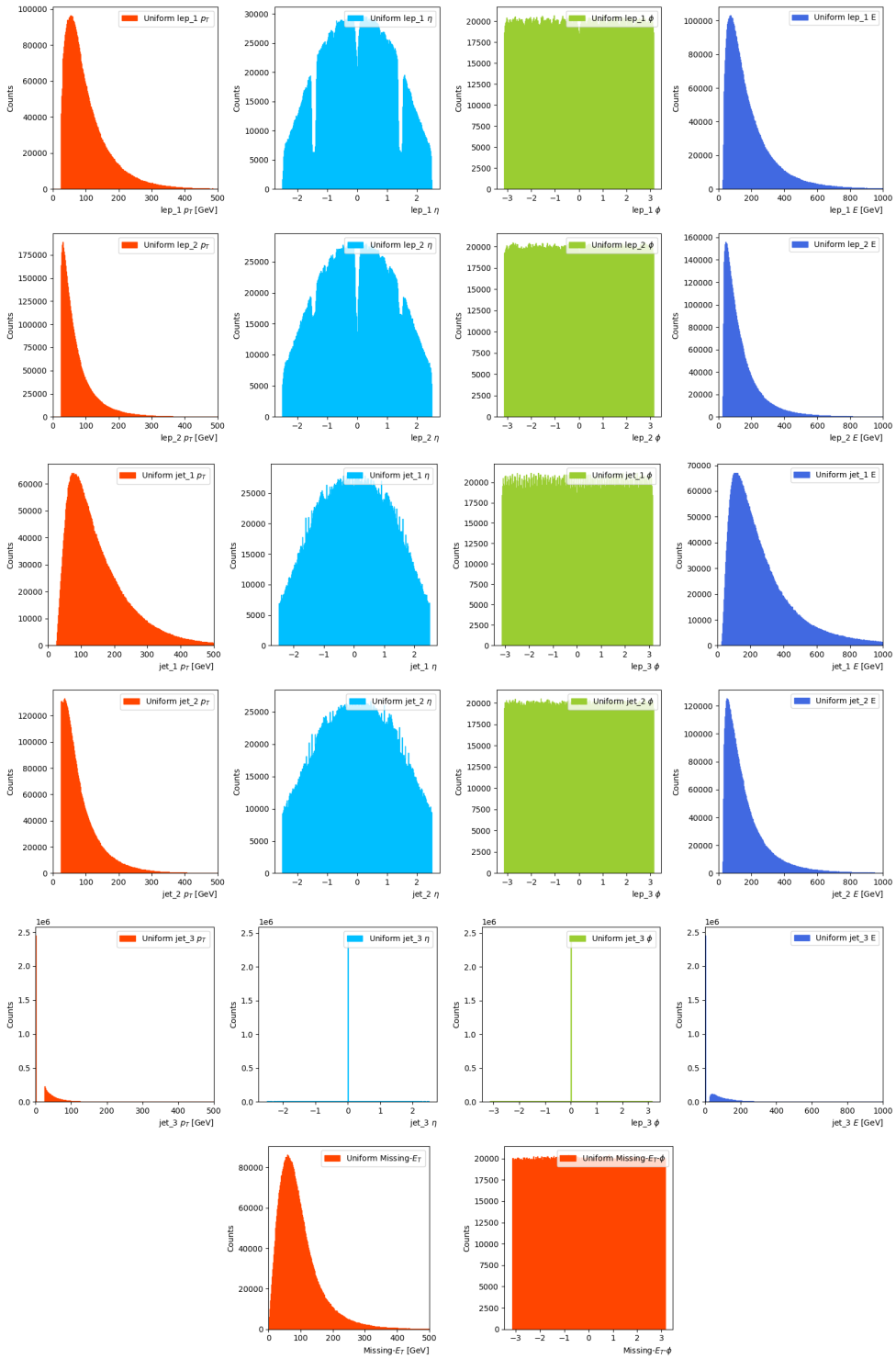

Figure A.2: Histograms of the 22 features that make up the "peak" dataset. In most events, only two jets are produced and the missing values, corresponding to the third jet, are filled with the 0 value.

Bibliography

[1] The ATLAS Collaboration. "The ATLAS Experiment at the CERN Large Hadron Collider", JINST 3 (2008) S08003.

[2] The CMS Collaboration, "The CMS Experiment at the CERN LHC", JINST 3 (2008) S08004.

[3] The ATLAS Collaboration. "Electron reconstruction and identification in the ATLAS experiment using the 2015 and 2016 LHC proton–proton collision data at $\sqrt{s}$=13 TeV", Eur. Phys. J. C 79, 639 (2019).

[4] The ATLAS Collaboration. "ATLAS b-jet identification performance and efficiency measurement with $t\bar{t}$ events in pp collisions at $\sqrt{s}$=13 TeV", Eur. Phys. J. C 79, 970 (2019).

[5] The ATLAS Collaboration. "Search for the standard model Higgs boson produced in association with top quarks and decaying into a $b\bar{b}$ pair in pp collisions at $\sqrt{s}$ = 13 TeV with the ATLAS detector", Phys. Rev. D 97, 072016 (2018).

[6] The ATLAS Collaboration. "Observation of the associated production of a top quark and a Z boson in pp collisions at $\sqrt{s}$ = 13 TeV with the ATLAS detector", JHEP volume 2020, Article number: 124 (2020).

[7] The CMS Collaboration. "Performance of photon reconstruction and identification with the CMS detector in proton-proton collisions at $\sqrt{(}s) = 8$ TeV", JINST 10 (2015) P08010.

[8] The ATLAS Collaboration. "Measurement of the tau lepton reconstruction and identification performance in the ATLAS experiment using pp collisions at $\sqrt{s}$ = 13 TeV." ATLAS-CONF-2017-029 (2017).

[9] The CMS Collaboration. "Observation of Higgs boson decay to bottom quarks", Phys. Rev. Lett. 121, 121801 (2018).

[10] M. Negri. "Machine Learning for di-tau invariant mass reconstruction in ATLAS". Thesis in Physics, Università degli Studi di Milano (2019). Available at: http://www.infn.it/thesis/PDF/getfile.php?filename=13671-Negri-triennale.pdf

[11] "Dark Energy, Dark Matter", NASA Science: Astrophysics. 5 June 2015.

[12] M. Kobayashi and T. Maskawa. "CP Violation in the Renormalizable Theory of Weak Interaction", Prog. Theor. Phys. 49, 652 (1973).

[13] K. Nakamura and Particle Data Group. "Review of particle physics", J. Phys. G 37, 075021 (2010).

[14] B. Pontecorvo. "Neutrino experiments and the problem of conservation of leptonic charge", Sov. Phys. JETP 26, 984 (1968); Z. Maki, M. Nakagawa, and S. Sakata. "Remarks on the Unified Model of Elementary Particles", Prog. Theor. Phys. 28, 870 (1962).

[15] Particle Data Group. "Review of Particle Physics", Chin. Phys. C 40 (2016), no. 10 100001.

[16] P. W. Higgs. "Broken symmetries, massless particles and gauge fields", Phys. Lett. 12, 132 (1964), Phys. Rev. Lett. 13, 508 (1964), Phys. Rev. 145, 1156 (1966).

[17] Particle Data Group. "Review of particle physics", Phys. Rev. D 98, 030001 (2018).

[18] CDF Collaboration. "Observation of top quark production in $p\bar{p}$ collisions", Phys. Rev. Lett. 74 (1995).

[19] D0 Collaboration. "Observation of the top quark", Phys. Rev. Lett. 74 (1995).

[20] J. C. Collins, D. E. Soper and G. Sterman. "Heavy particle production in high-energy hadron collisions", Nucl. Phys. B 263, 37 (1986).

[21] Summary plots from the ATLAS top physics group (2015). Available at https://atlas. web.cern.ch/Atlas/GROUPS/PHYSICS/Combined SummaryPlots/TOP/

[22] LHCTopWG - LHC Top Physics Working Group (2016). Available at http://twiki.cern.ch/twiki/bin/view/LHCPhysics/LHCTopWG/

[23] T. G. Rizzo. "Z' Phenomenology and the LHC", SLAC-PUB-12129, arXiv:hep-ph/0610104v1 (2006).

[24] The ATLAS Collaboration. "Search for heavy particles decaying into top-quark pairs using lepton-plus-jets events in proton–proton collisions at $\sqrt{s}$=13 TeV with the ATLAS detector", Eur. Phys. J. C 78, 565 (2018).

[25] LHC Guide, tech. rep. (2017). Available: https://cds.cern.ch/record/2255762.

[26] The LHCb Collaboration. "The LHCb Detector at the LHC", JINST 3 (2008) S08005.

[27] The ALICE Collaboration. "The ALICE experiment at the CERN LHC", JINST 3 (2008) S08002.

[28] The LHCf Collaboration. "The LHCf detector at the CERN Large Hadron Collider", JINST 3 (2008) S08006.

[29] The TOTEM Collaboration. "The TOTEM experiment at the CERN Large Hadron Collider", JINST 3 (2008) S08007.

[30] The MoEDAL Collaboration. "Technical Design Report of the MoEDAL Experiment", CERN-LHCC-2009-006 (2009).

[31] M. Cacciari et al. "The anti-k_t jet clustering algorithm", JHEP 04 (2008).

[32] McCulloch, W. S., Pitts, W. "A logical calculus of the ideas immanent in nervous activity", Bulletin of Mathematical Biophysics 5, 115–133 (1943).

[33] Rosenblatt, Frank (1957). "The Perceptron—a perceiving and recognizing automaton", Report 85-460-1. Cornell Aeronautical Laboratory.

[34] Cybenko, G. "Approximation by superpositions of a sigmoidal function", Mathematics of Control, Signals, and Systems. 2 (4): 303–314 (1989).

[35] S. Hochreiter, Y. Bengio, and P. Frasconi, "Gradient Flow in Recurrent Nets: the Difficulty of Learning Long-Term Dependencies", in "Field Guide to Dynamical Recurrent Networks", J. Kolen and S. Kremer, eds. IEEE Press (2001).

[36] X. Glorot, A. Bordes and Y. Bengio. "Deep Sparse Rectifier Neural Networks", Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings 15:315-323 (2011).

[37] M. Miljanovic. "Comparative analysis of Recurrent and Finite Impulse Response Neural Networks in Time Series Prediction", Indian Journal of Computer and Engineering. 3 (2011).

[38] M. Minsky 1927-2016., Perceptrons; an introduction to computational geometry" [2d print. with corrections by the authors] MIT Press (1972).

[39] H. Mhaskar, Q. Liao, and T. Poggio. "Learning Real and Boolean Functions: When Is Deep Better Than Shallow", Center for Brains, Minds and Machines, arXiv: 1603.00988 (2016).

[40] Y. LeCun, Y. Bengio, G. E. Hinton, "Deep learning", Nature 521, 436–444 (2015).

[41] A. Krizhevsky, I. Sutskever and G. E. Hinton. "Imagenet classification with deep convolutional neural networks", Communications of the ACM, 60(6):84–90 (2017).

[42] T. Mikolov, A. Deoras, D. Povey, L. Burget, and J. Cernocky. "Strategies for training large scale neural network language models", 2011 IEEE Workshop on Automatic Speech Recognition and Understanding, 12 (2011).

[43] T. Karras, T. Aila, S. Laine and J Lehtinen. "Progressive growing of gans for improved quality, stability, and variation", CoRR (2017).

[44] J. Ma, R. P. Sheridan, A. Liaw, G. E. Dahl, and V. Svetnik. "Deep neural nets as a method for quantitative structure-activity relationships", Journal of Chemical Information and Modeling, 55(2):263–274 (2015).

[45] P. Baldi, P. Sadowski and D. Whiteson. "Searching for Exotic Particles in High-Energy Physics with Deep Learning", Nature Commun. 5, 4308 (2014).

[46] D. C. Ciresan, U. Meier, J. Masci, L. M. Gambardella, J. Schmidhuber. "Flexible, High Performance Convolutional Neural Networks for Image Classification", Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence-Volume Volume Two. 2: 1237–1242 (2011).

[47] R. Di Sipio, M. Faucci Giannelli, S. Ketabchi Haghighat and S. Palazzo, "DijetGAN: A Generative-Adversarial Network Approach for the Simulation of QCD Dijet Events at the LHC", arXiv: 1903.02433 [hep-ex] (2019).

[48] D. P. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization", arXiv:1412.6980 (2014).

[49] M. D. Zeiler. "Adadelta: An Adaptive Learning Rate Method", arXiv:1212.5701 (2012).

[50] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization", Journal of Machine Learning Research 12 no. Jul (2011).

[51] G. Hinton, N. Srivastava and K. Swersky. "Neural Networks for Machine Learning - Lecture 6e, RMSProp: Divide the gradient by a running average of its recent magnitude" (2014). Available at: http://www.cs.toronto.edu/ hinton/coursera/lecture6/lec6.pdf

[52] L. Bottou, F. E. Curtis and J. Nocedal, "Optimization Methods for Large-Scale Machine Learning", arXiv:1606.04838 [stat.ML] (2018).

[53] I. Goodfellow, Y. Bengio and A. Courville. "Back-Propagation and Other Differentiation Algorithms in Deep Learning", MIT Press (2016).

[54] M. A. Nielsen. "How the backpropagation algorithm works", in "Neural Networks and Deep Learning", San Francisco, CA: Determination press (2015).

[55] G. Van Rossum and F. L. Drake Jr. "Python reference manual", Centrum voor Wiskunde en Informatica Amsterdam (1995).

[56] The pandas development team. Zenodo. doi 10.5281/zenodo.3509134. url https://doi.org/10.5281/zenodo.3509134

[57] C. R. Harris et al. "Array programming with NumPy", Nature 585(7825), 357-362 (2020).

[58] J. D. Hunter. "Matplotlib: A 2D Graphics Environment", Computing in Science and Engineering, vol. 9, no. 3, pp. 90-95 (2007).

[59] F. Chollet et al. "Keras", XH Lu et al./Application of Machine Learning and Grocery Transaction Data (2015). Available at: https://keras.io

[60] A. Martin et al. "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems", arXiv preprint arXiv:1603.04467 (2015). Available at: http://tensorflow.org/. Software available from https://www.tensorflow.org/.

[61] S. Agostinelli et al. "Geant4 - A Simulation Toolkit", Nucl. Instrum. Meth. A 506 (2003).

[62] S. Alioli, P. Nason, C. Oleari and E. Re. "A general framework for implementing NLO calculations in shower Monte Carlo programs: the POWHEG BOX", JHEP 1006 (2010) 043.

[63] T. Sjostrand, et al. "An introduction to PYTHIA 8.2", Comput. Phys. Commun. 191 (2015).

[64] The D0 Collaboration. "Precise measurement of the top quark mass in dilepton decays using optimized neutrino weighting", Physics Letters B, Volume 752 (2016).

[65] The ATLAS Collaboration, "Measurements of top-quark pair spin correlations in the $e\mu$ channel at $\sqrt{s} = 13$ TeV using pp collisions in the ATLAS detector", Eur. Phys. J. C 80 (2020) 754.