# Machine learning for jet energy unfolding in ttbar analysis

*DESY Summer Student Programme, 2019*

Luiza Adelina Ciucu

*University of Geneva, Switzerland*

Supervisor
Thorsten Khul, Yichen Li

September 5, 2019

**Abstract**

## Contents

# 1 Introduction

At this moment, The Large Hadron Collider (LHC) which is situated at CERN is the most powerful proton-proton collider in the world, as illustrated in Figure **??**. After Tevatron and LEP (Large Electron Collider) era, a new machine was needed for new discoveries in particle physics. The LHC was designed to achieve a center of mass energy $\sqrt{14}$ TeV. Two of the biggest goals of the LHC is to study the Standard Model and to test its validity.

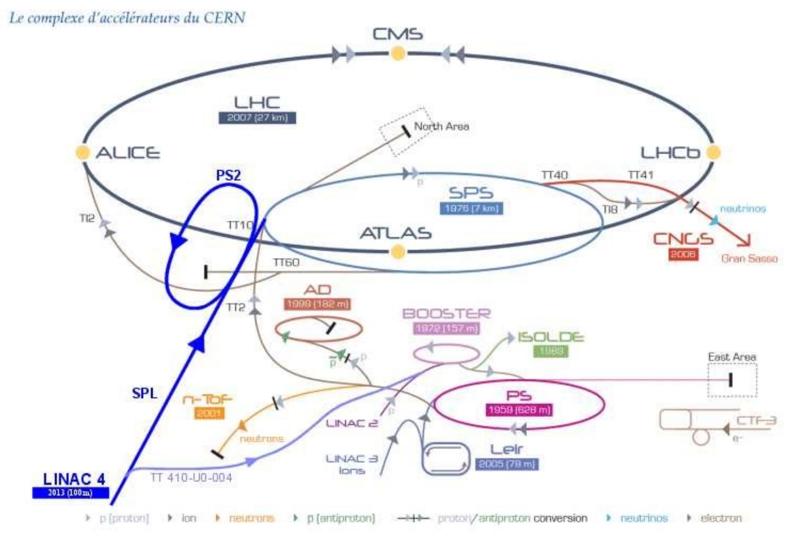

Figure 1: The LHC Accelerator Complex System.

This analysis is based on ATLAS (A Toroidal LHC ApparatuS) experimental data measurements. ATLAS is one of the four LHC's detectors, being the largest machine ever built. It is a general-purpose particle physics, run by an international collaboration [**?**].

The Standard Model is a theory that classifies the elementary particle (fermions and bosons) and describes three fundamental forces. It has 12 fermions that can be divided in two categories, quarks and leptons and they are the constituents of the matter. There are five bosons, the gluon which carry the strong force, photon responsible for electromagnetic force, W and Z bosons carrying the weak force. The most important boson is the Higgs, the responsible for the mass of all the elementary particles.

Top quark is an interesting particle to study for several reasons: is the heaviest particle from the Standard Model (mt=173.3 GeV/c2) implies a high Yukawa coupling constant. Measuring the top-quark properties is the key to test the validity of the Standard Model.The . Another reason will be that decay very quickly (tau 10 -25 s). And if we find deviations in the top quark properties from the Standard Model predictions (e.g different cross-sections) implies the existence of new physics Beyond the Standard Model which is a very big purpose. Its evidence was discovered in March 1994 at CDF experiment at FermiLab in pp collisions at $s = \sqrt{1.8}$ TeV. Top quarks are produced in pairs by strong interactions by quark-antiquark annihilation 10%, as gluon gluon fusion 90%. The LHC has very high luminosity and the top quark are well understood. So because of this we can stay that the LHC is a top quark factory.

At leading order the pair production of tt pairs is described by the following Feynman diagrams. We are interested about dileptonic channel. Even it is presented only in 2% of the cases, it is advantageous because the background is very low comparing to others decay channel. The background comes in most of the cases from Z boson decay plus other jets.

## 2 Neural Networks

The goal of this project is to adapt a code example of machine learning unfolding (MLUnfolding) using toy data to use it for the jet energy reconstruction of ttbar e-mu analysis.

In this project one of the problems that we want to solve is the correction of detector smearing using Machine Learning. I have implemented this method using a sequential Neural Network. The unfolding correspond to an inversion of the migration matrix. Unfolding is the procedure to infer the truth data (what happened in reality in nature) from the reconstructed data (observed and measured in our experiment by our detector). The categorisation is a common problem for the modern Machine Learning methods. Possible methods could be boosted decision trees and Artificial Neural Networks. Neural Networks with various architectures can be tried for the unfolding problem.The unfolding involves several iterative steps of machine learning using a neural network training in TensorFlow via Keras, in Python. The NN takes as input the reconstructed values, and has as output the truth values.

The goal is to adapt this example to use real data coming from the jet pt distribution from the ttbar e-mu analysis. We use this .root flat tree [?]. This file contains both the reconstructed and truth jets, already matched. Meaning the i-th event from reco, corresponds to the i-th event from truth. Each jet collection collects jets already the jets ranked by pt. We take index 0 to consider the leading reco jet and the leading truth jet.

Given the leading jet pt as a continuous reco value, we want to find out in which jet pt bin (set by user, say 10 GeV or 20 GeV) does the truth jet falls. The input is a continuous value, but the output is a discrete value, that can take values from 0 to 49 if we consider 10 GeV bins from 0 to 500 GeV. Bin 0 contains jets with pt from 0 to 10 GeV. Jets with pt larger than 500 GeV have values set by hand to 499.999 GeV. It is equivalent to moving the overflow of a histogram bin to the last bin of the histogram. Given the output can take any value from 0 to 49, the problem we try to solve is a classification, and the possible labels are not only two (as in a simple signal to background classification, typically used in ATLAS), but a more complex one, with 50 labels. The NN will return the probability that for a given jet its pt falls in any of these bins. The total probability for all bins must be 1.0. This is ensured by the softmax activation function for the last layer of the NN. We then consider the bin with the largest probability as our choice by the NN. The predicted bin value can then be compared with the true bin value when making the plots of this project.

To optimize the NN performance, we take as input in fact not the jet pt, but the jet pt divided by the bin width. This is the jet pt bin as a real value. Making things more consistent with the output being the integer jet pt bin value for the truth.

Neural Networks are an example of Machine Learning. In this way the computers learn a solution for a problem without being explicitly programmed. The two main classes of ML are supervised and unsupervised. In this project I used supervised ML. ...... [?].

If we have the function Pi(y), a multidimensional highly non-linear, an efficient way to do this procedure is with an NN. This was inspired by the brain structure, which contains millions of neurone cells forming a network with electrochemical impulses passing between them. An artificial neural network formed by a number of interconnected artificial *neurons*, or "nodes" respects this architecture.

A network is formed by several ayers of nodes connected in series. Each node takes a weighted linear combination of the outputs from nodes of the previous layer, applies "activation function", then outputs the result do the next layer [?].

Using a "loss" function we can describe the difference between the predicted and real outputs, like the mean squared error between real and predicted. The weights associated with each node is modifies via an algorithm, and from here we can deduce that the loss function decreases and training increases.

We know from The *Universal Approximation Theorem* that a neural network with one *hidden layer* of nodes between input and output can in principle approximate any N-dimensional function to an arbitrary degree of accuracy, given a sufficiently large (though finite) number of nodes.

In practice it is more suitable to use multiple hidden layers connected in series [?].

## 3 The method

My first step was to study a code example using some toy data in a JupyterNotebook [?]. At the beginning I ran the code on a cloud (Swan).

The next step was to run it on `lxplus`. For this it was needed to run after `ssh`-ing to a `lxplus` machine a singularity command [?] and then to my own laptop.

To train the neural network it was needed to use the Python library Keras [?] and for backend TensorFlow [?]. My method of sampling was implemented using numpy [?].

## References

[1] The ATLAS experiment, `https://atlas.cern`

[2] The ttbar e-mu ROOT file used `/afs/cern.ch/user/l/lciucu/public/data/MLUnfolding/user.yili.18448069._000001.output.sync.root`

[3] Joshua Bendavid, *Efficient Monte Carlo Integration Using Boosted Decision Trees and Generative Deep Neural Networks*, 2017

[4] Andrew Ng, *Machine Learning. Coursera online course*, URLhttps://www.coursera.org/learn/machinelearning

[5] Alexander Glazov, *Machine learning as an instrument for dana unfolding*, **arXiv**: 1712.01814v1, http://inspirehep.net/record/1641082

[6] The singularity command to run the ML environment,
`singularity exec '/cvmfs/unpacked.cern.ch/registry.hub.docker.com/atlasml/ml-base:latest'bash`

[7] The keras machine learning library, https://keras.io

[8] The TensorFlow machine learning library, https://www.tensforflow.org

[9] The Numerical Python library, https://www.numpy.org

[10] *Measurement of jet activity produced in top-quark events with an electron, a muon and two b-tagged jets in the final state in pp collision at s=13 TeV with the ATLAS detector*, **arXiv**: 1610.09978v2, https://arxiv.org/pdf/1610.09978.pdf