Machine Learning in Particle Physics

PHY 376 : Design Project

Project Report

Sharad Chitlangia

ID No.: 2017A8PS0472G

30th April 2019

Acknowledgements

I would like to thank Dr. Kinjal Banerjee for giving me the opportunity to work on this wonderful project, and for providing constant guidance throughout the project duration. The project involved some concepts from particle physics which I didn’t understand until I was given direction by him. He also encouraged me to make submissions for the competition which led to a 6th rank in the competition.

Contents

[1] The competition page is at https://competitions.codalab.org/competitions/20112. My final rank was **6th** on the official challenge and **1st** in the private competition. This can be found under the 'Development' tab in Results tab.
[2] The code for all the algorithms can be found at https://github.com/Sharad24/Particle-Track-Reconstruction

Chapter 1

Introduction

Particle Physics is a vast field in Physics where elementary particles, their interactions, and the theories are studied in great detail. Experimental Particle Physics attempts to verify the results that theoretical Particle Physics predicts, through very advanced and expensive technology. The most famous of these is the Large Hadron Collider at CERN, Geneva, where the major discoveries of particle physics occurred.

Machine Learning is a very important and upcoming field of Computer Science which uses complex algorithms to teach computers how to learn patterns in data and predict outcomes for events based on some experience. This area of computer science is extremely powerful and is nowadays used everywhere, like in recommender systems, handwriting recognition, weather prediction, gadget customization and so on. There are several subfields of machine learning like deep learning, reinforcement learning etc., each introduced to solve different kinds of problems.

This report is an effort to understand the use of Machine Learning techniques in Particle Physics. Experimental Particle Physics annually analyses several terabytes of data, and machine learning plays a big role in this analysis. To understand this better, this report looks at how machine learning helps improve predictions and saves a lot of effort from the physics community annually to devise smarter and smarter ways to get the results they want.

Chapter 2

Particle Physics in Colliders

Colliders like the Large Hadron Collider at CERN, Tevatron at Fermilab, DESY at Hamburg all aim to solve unanswered questions in Physics by extracting useful information from collisions of beams of particles at great energies. These collisions give rise to more particles and radiation that we may already have guessed, or ones we do not expect to find. The goal of Particle physics in colliders today, is to investigate processes that will help find new particles or new interactions between particles, to solve some mysteries plaguing Physics.

2.1 What happens in Colliders

Theorists work on hypothesizing and predicting what interactions particles must have with each other, and what might be the outcome of collisions between particles. This must then be verified in accelerators where particles are actually collided with each other, using very long accelerator tubes, so that at very high energies, colliding beams may bring the particles close enough to experience interactions that are not often seen in daily life, like weak and strong interactions. For example, ATLAS is one of the accelerators at CERN that collides protons with each other. For the longest time, it was not confirmed that protons had substructure, but by colliding very high speed electrons with protons, and later protons with protons, many elementary particles like quarks, Higgs Boson were discovered. These particles are extremely unstable and are not directly detected, analysis of data from colliders about the final products helps to back track and tell which particles might have decayed to them.

The process of colliding an electron or muon with a proton to probe its constituents like quarks, gluons and other particles is called *deep inelastic scattering*. Quarks and gluons interact with each other to produce new quarks, and gluons interact with each other to produce particles, some of which may be unknown and can help probe new physics. The final products have hit the detectors and their momenta, velocity, charge and angles are measured. This helps to identify which particles have been produced, and given their momenta and angular distribution, which particles might have decayed to them.

The *3 dimensional momentum* is measured by the first layer of the detector, called an inner tracker. It has very strong magnets that create a magnetic field strong enough to curve the paths of particles based on their momenta. The direction of the curve (left or rightward bending) is decided by the *charge* of the particle. Since these particles are moving relativistically, high momentum particles curve less and low momentum particles curve more.

Particle identification happens by a combination of velocity and momentum measurements. While momentum is measured in a tracking chamber, *velocity* is measured in various ways. In a *Time of Flight* detector, the time required to travel from the interaction point to the detector, or the time required to travel between two time of flight detectors is measured, and velocity is calculated taking into account other resistive forces. Time of flight decreases as particles move closer to speed of light.

2.2 Signal and Background

Reactions in which the intermediate products are those of interest to us, we call signal. The other reactions that produce the same final particles are called background. The physics community devises several features to identify the difference between these two reactions, simply by observations made in detectors. Some features are directly measured- called low-level features, and some features are linear combinations, or derivable from the low level features, called high-level features.

2.3 Challenge description

In the LHC, proton bunches (beams) circulates and collide at high energy. Each collision produces a firework of new particles. To identify the types and measure the kinematic properties of these particles, a complex apparatus, the detector bends them in an externally applied magnetic field and records the small energy deposited by the particles when they impact well-defined locations in the detector.

The tracking problem refers to reconstructing the trajectories from the information recorded by the detector. Given the coordinates of the impact of particles on the detector (3D points), the problem is to “connect the dots” or rather the points, i.e. return all sets of points belonging to alleged particles trajectories.

Chapter 3

How can Machine Learning help?

As the detectors at LHC, Cern become more stronger, the amount of data generated becomes higher. Due to some natural errors like Noise, Lack of constant magnetic fields and secondary trajectories, combinatorial methods cannot be applied. Machine Learning models are robust to small noise and are great at detecting patterns in a large amount of data.

3.1 Data

Event hits

- hit id: a numerical hit identifier, unique inside the event.
- x, y, z : the reconstructed position (x, y, z)
- volume id,layer id, module id: the module where the hit was read out by using the identification scheme

Event Particles

- particle id: a unique identifier of a particle inside the event.
- vx,vy,vz: the coordinates (in millimeters) of the particle vertex , ie the origin of the trajectory.
- px,py,pz: initial momentum (in GeV/c) at particle vertex
- q : the charge as multiple of the absolute electron charge
- nhits: number of hits generated by this particle

Event truth file

The truth file contains the mapping between hits and generating particles, together with the true particle state at each measured hit. Each entry maps one hit to one particle.

- hit id: numerical identifier of the hit as defined in the hits file.

- particle id: numerical identifier of the particle generating this hit as defined in the particles file.
- tx, ty, tz: true hit position (in mm) at the intersection with the module. Remember that the true hit position is the intersection of the trajectory with the module mid-plane.
- tpx, tpy, tpz true particle momentum (in GeV/c) in the global coordinate system at the intersection point. The corresponding unit vector is tangent to the particle trajectory.
- weight: per-hit weight used by the scoring function; note that the total sum of weights within one event equals to one.

Event hit cells

- hit id: numerical identifier of the hit as defined in the hits file.
- ch0, ch1: cell identifier/coordinates unique with one module, encoded as in figure 6. Depending on the detector type only one of the channel identifiers may be valid.
- value: Signal value information, e.g. how much charge a particle has deposited.

3.2 Approach and Preprocessing

Approach

I developed a general framework to be followed for training algorithms. This framework was largely motivated by some of the algorithms exposed in the first phase (the Accuracy phase) of the competition on Kaggle. The framework is described below:

1. Selection of Promising Pairs of Hits
2. Extension of Pairs to Triples
3. Track Extension from Triples
4. Addition of duplicate Hits
5. Removal of Overlapping tracks

Preprocessing

The input features considered for pair classification were the hit locations, cell and signal values. The hit location is in cartesian geometry. The cartesian coordinates were converted into cylindrical as it simplifies the task for classification for the classifier. This is primarily because a helix in cylindrical coordinates is like a straight line in cartesian coordinates.

3.3 Algorithms Tested

The task is develop a method for identification of Promising pairs and triplets. This is plausible as two hits cannot be in two very different sections of the detector if they are part of a trajectory. A multitude of algorithms were tested for pair classification. They are described below:

- Logistic Regression: In its basic form uses a logistic function to model a binary dependent variable. The logistic function is described as:

  $$f(x) = \frac{1}{1+e^{-x}}$$

  The model is described as:

  $$f(x) = f_1\left(\sum_{i=0}^{n} w_i \cdot x_i\right)$$

  where,

  $f1(.)$ is the logistic function as described above.

  The test accuracy achieved was 75.3%.

- Artificial Neural Network: An ANN is based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit a signal from one artificial neuron to another. An artificial neuron that receives a signal can process it and then signal additional artificial neurons connected to it. Each node represents what is called a perceptron. The perceptron equation is described as:

$$f(x) = w \cdot x + b$$

where,

w represents the weight for that node

b represents the bias for the node

Essentially a neural network is a combination of such perceptrons, which makes it such a powerful function approximator. The ANN is trained using the Backpropagation algorithm. For the purpose of this challenge a network comprising of five layers was used. The number of neurons in each layer, in sequence, was 4000, 2000, 2000, 1000, 1 (Figure 1).

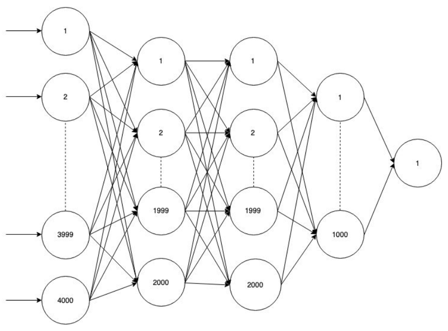

Figure 1

The test accuracy achieved with this Neural Network is 84.3 % for pair classification. Overall test track reconstruction accuracy was 80.4 %.

- DBSCAN Clustering: It is a density-based clustering non-parametric algorithm: given a set of points in some space, it groups together points that are closely packed together (points with many nearby neighbors), marking as outliers points that lie alone in low-density regions (whose nearest neighbors are too far away). DBSCAN is one of the most common clustering algorithms and also most cited in scientific literature. Clustering was performed on input data which gave a test accuracy of 53.3 %. Overall track reconstruction accuracy was 46 %.

- Random Forest: They are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class. The prediction is made by a majority vote among the trees. A random forest consisting of 500 trees with a depth of 10 was trained on particular event data. The resultant trained Random Forest gave an accuracy of 94.6 % ontest data.

- Gradient Boosted classifier: It is a machine learning technique for classification problems, which produces a prediction model in the form of an ensemble of weak prediction models like decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function. It re-trains the model on false positives and false negatives so as to make the model learn better. Highest accuracy was achieved with this classifier. The test accuracy was 98.1%. Overall accuracy was 90%. This currently stands ***first*** on the private CERN leaderboard.

3.4 Benchmarks

Algorithm	Train Accuracy	Test Accuracy	Train time	Total Inference time[2]	Overall Track Reconstruction Score
DBSCAN	56%	53.3%	4 hours	ε [1]	46%
ANN	86%	84.3%	4 days	6 days	80.4%
Logistic Regression	76%	75.3%	30 s	ε [1]	67%
Random Forest	95.3%	94.6%	30 minutes	10 minutes	87.2%
Gradient Boosting Classifier	98.1%	97.5%	30 minutes	10 minutes	90%

[2] The total inference time includes the time for pair prediction and the time for trajectory reconstruction

[1] ε signifies negligible time. Less than a second

Chapter 4

Outlook

This report discusses the applications of Machine Learning and Deep Learning to one of the tasks at CERN’s experiments. A faster algorithm can be built in C++ for faster inference using CERN’s ROOT library. An algorithm which also organises the event data in a directed Acyclic Graph should work better. Another easy extension would be to train an algorithm to predict triples from pairs since as of now I have just extended the trajectory using 3d geometry.

Appendix A

The reports for the top algorithms can be found in the drive folder:

https://drive.google.com/open?id=1cnp8oHTaS8Nx33cKPwCzsPea4wAloKd7

Bibliography

[1] The TrackML challenge https://sites.google.com/site/trackmlparticle/

[2] The Discussions at https://www.kaggle.com/c/trackml-particle-identification/discussion/

[3] Pytorch - Deep Learning Library: https://pytorch.org/

[4] Scikit-Learn, machine learning package for python, https://scikit-learn.org/

[5] S. Sekhar. The Higgs Boson Machine Learning Challenge