TrackML Particle Tracking Challenge

Portia Allen

Connor Caldwell

Arnaud Filliat

Johnny Murphy

12 October 2020

Abstract

Overview and Related Work

Subatomic particle detection has become critical to deepening our understanding of what makes up the universe. Scientists at CERN are colliding protons using the Large Hadron Collider (LHC), and observing these small scale, big-bang-esque collisions to learn more about fundamental particles. However, as observation techniques continue to increase in quality, so does the quantity of data that documents a single collision. The purpose of this project is to use machine learning algorithms to accurately categorize high energy particle collisions, and then predict new particles.

This is possible through a reconstruction of the particle events from the LHC. Using the dataset provided by CERN scientists, it is possible to break down confirmed collision events and categorize them by particle type, and then use that algorithm with unknown collision data to predict the resultant particles. Each event contains a lot of data, such as the coordinates of the collision and the particle charges, and then the trajectories of the post-collision particles. Therefore, it is critical to have a higher degree of understanding of what the data represents, in order to build accurate models and correctly interpret the training data, as well as categorize unknown data using the same parameters.

By creating this machine learning algorithm, we can more efficiently and effectively track and categorize high energy particle collisions. We can gain a better understanding of the particles, how they collide, and what particles they break into after the collision. Additionally, several members of our group have a deep background in physics, so this dataset is of particular interest. Our personal connection to the discipline can be more motivating for the completion of this project. The application of data science to this project is extremely straightforward – given some dataset, we have to parse through it, train a machine learning algorithm with some of it, test it using the rest of the data, and then refining our model to be the most accurate it can be. Through this process, we can develop a better understanding and ability to apply data science/machine learning concepts to real-world data, while also gaining a deeper appreciation for and understanding of current particle physics research.

It should be noted that this dataset was used as part of a machine learning algorithm competition in 2018. The goal was to write an algorithm with the highest accuracy, and then highest throughput (speed relative to accuracy). Several people who placed highly in the competition have posted their answers and solution process online, so that other people can understand how to approach and execute an algorithm like this. Therefore, our group can make a solid attempt at it with our more limited data science skills, and then compare to those who have more experience in the field; from this, we can see what we could improve, as well as gain more insight into applied data science.

Examples of the other solutions

Data Acquisition

The dataset that we intend to work with is the Particle Tracking dataset from the Large Hadron Collider at CERN. Since this data was made publicly available by CERN for the TrackML Particle Tracking Challenge in 2018, there are no restrictions on our use or sharing of the data.

We are given a test dataset with 125 collision events upon which to evaluate our model; a training dataset, 8850 events split into five different files, as well as a sample set (the first 100 events from the training dataset); and some informational files about the geometry of the detectors. These are all provided by CERN in zipped .csv files, that can be downloaded and then read into a Pandas DataFrame for ease of access and manipulability. Additionally, CERN provides a trackml python library in GitHub to simplify some of the data handling, which will most likely be used extensively to aid in data visualization and processing.

Each event has four associated files, containing the hits, the hit cells, the particles, and the ground truth for the event. The hit files contain the identification numbers for the hit itself and the detector group/layer/module location of the hit, as well as the x-y-z coordinates for the hit. The truth files have the hit identification number, the particle identification number, the true hit location and particle momentum, and the weight of the hit (for the scoring metric). The particle files contain the particle ID number, the particle type, initial position and momentum, charge of the particle, and number of hits from this particle. The hit cell files contain the hit ID number, how much charge a particle has deposited on the cell, and the cell coordinates. The cells are the smallest positional identifier on the detectors, and can be used to more accurately track association between hits and particles.

The geometry of the detectors is important information to know because the detector is built from concentric silicon slabs that have been subdivided several times. The largest groups are volumes, subdivided into layers, which are then divided into modules, which are made up of cells. Each of those have ID numbers except the cells, which have a gridded identification system. Each module has a different local position and orientation, so a transformation must be made between the local coordinates of the hit and the global coordinates of the hit to get the actual path of the particle.

Preprocessing

Below is a graph of the X values compared to the Y values in one event. As you can see this looks like a sphere which is expected since the CERN is a cylindrical model, and this allows for us to view the points in which a data point can be constrained in, and then shows that there are a couple of rings where it looks like some events where not recorded, which could help us in our training in order to solve some issues that could be a cause for concern in this model.

In this visualization below, it is shown that the X Values compared with the y values are in a cylindrical shape, which is expected. This shows that there are certain heights that were omitted in the measurements, which could be a cause for concern when trying to classify particles within the models.

This histogram below shows the values of this hit as a histogram with about 20 bins. This shows that the data is very skewed to the right, which means a certain particle could be much heavier, classifying it as something else, such as the Higgs Boson. The data on the left can be abstracted to particles such as beta particles, which are lighter and much more abundant, same with Z particles as well.

The graph below displays the particle’s velocity on an X and Y plane, this displays that the particles were going very rapidly and randomly in a circle. With this, this allows for us to analyze the total mass of the particle and being able to classify since the momentum will be less with higher mass particles.

Model Selection

The main algorithm we want to use is a machine learning technique using support vector machines. These will allow us to be able to predict where new data points will be using previous data points, which is perfect for tracking a 3D particle. Since the particles we are using are in a multidimensional space, the best way to use the SVM software is to utilize the kernel function, allowing for nonlinear models to be produced.

In this case the nonlinear models would be most useful in the X and Y position of the particles, allowing us to block off different groups and relate them to other features, such as the X and Y velocity. This will hopefully allow for more accurate predictions while maintaining data integrity.

SciKit-Learn has multiple functions that can be utilized for SVM machine learning, including the kernel function as mentioned prior, as well as a C variable that allows for flexibility. This prevents overfitting or underfitting allowing a unique fit for each model. The leniency in our function will have to be tested, since the particles are semi-random and are hard to get a locked down number on. It will likely be in the range of C = .1-1 area since there are so many data points.

Due to the random nature of particles and how they move another possible algorithm to use would be an unsupervised learning algorithm. We were thinking of using k-means as well to split the clustering and possibly look at reducing dimension to only the most important features. These features could include the particles’ initial velocities and their charge. This would enable a more “learned” approach which could be significant but due to the large amount of data may be impractical because of the time to run.

Results and Evaluation

Ethics

Future Work

Works Cited