Machine Learning

Potential Approaches: LHC Trajectory Tracking

Barcelona, April 14th 2020

Jake Watson (1 person)

The problem is essentially connecting the dots: hits are arranged in several concentric rings. These hits are caused by secondary particles, which result from a set of initial particles. The problem is therefore to group the hits into tracks, each resulting from a single initial particle. The dataset is composed of the recorded hits, their ground truth classification of the hits into tracks with their initial particles, and the parameters of the input particles. To allow the use of supervised machine learning for the problem, we must find a way to turn individual hits into tracks, so that they can be compared to the ground truth.

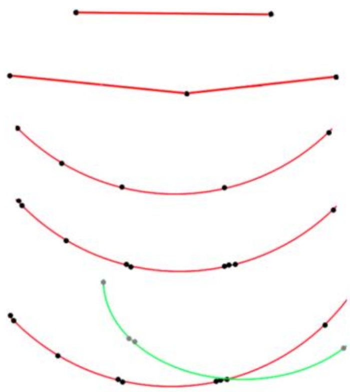

Figure 1: illustrating the steps of the trajectory classifier.

Iterative Classifier Approach

One approach is to iteratively build each track. This can be done by combinatorially generating a set of triplets, or seeds, in the first and second rings, and then extrapolating from these seeds. The potential combinations of hits into triplets could be very large, so a binary classifier can be used to reduce the search space.

By following the above procedure, a set of potential tracks is generated for a single initial hit in the central ring. Each central hit can only correspond to one track, and so one track is chosen using a classifier, using properties of the track such as number of hits assigned to it, the summed distance between each hit and the extrapolation, and so on.

Deep Learning Approach

Another approach is to train a neural network on a set of labelled tracks. The inputs to the network are the coordinates of the innermost hits, and the outputs are the predicted positions of hits in the outer layers. The network is learning to estimate the parameters of the helixes in the training set.

To use this network to generate tracks, we need to use a seeding method to generate potential tracks in the inner layers of hits, potentially using a similar combinatorial method combined with a classifier to determine 'good' potential tracks. Using the potential inner tracks as input, a set of potential full paths are generated.

These are then associated to actual hits in the outer layers, by adding those hits that are closest to the path prediction. This method differs from the iterative method, as it generates a path from an initial seed in one go, rather than building the track by adding hits one by one.

Combinatorial Deep Learning Approach

A third approach combines combinatorial approaches with deep learning methods. To generate a set of potential tracks, all possible pairs of hits are generated, with a set of features to describe the pair. These features are used to train a neural network to predict the likelihood of the hits in each pair belonging to the same track.

The pairs are then extended by adding other pairs that maximise the summed probability of belonging to the same track. Only those pairs that pass a set of geometric conditions for possible tracks are combined. This procedure is repeated until a full track is generated.

Final Approach

The approaches above summarise some of the best-scoring approaches in the Kaggle competition for this dataset[1]. I believe that there is potential here to combine and build upon these approaches, taking the best of each. However, I have not yet settled on exactly which approach I will take, and need to read further.

[1] 'The Tracking Machine Learning challenge : Accuracy phase', Amrouche. S, Basara. L, et.al., , https://arxiv.org/pdf/1904.06778.pdf