

A novel approach to assist fast tracking at the ATLAS trigger

A. ZAIO ON BEHALF OF THE ATLAS COLLABORATION

INFN & Università di Genova - Genova, Italy

Summary. — The track reconstruction process within the trigger system of the ATLAS experiment shows a processing time which increases significantly as a function of the average pile-up amount of the pp collisions, so that the future upgrade to the High-Luminosity LHC (HL-LHC), with way higher levels of pile-up, will imply a considerable growth of the computational cost for the current trigger algorithms. To face this issue, an innovative machine-learning-based technique to assist tracking by filtering out background hits is presented and characterized. The algorithm is based on a Convolutional Neural Network architecture and is trained and tested using an independently developed toy event generator.

1

2 1. – Introduction: online tracking in ATLAS

3 While the Level 1 (L1) trigger of the ATLAS experiment [1] operates using the
4 hardware information coming from the calorimeters and the muon spectrometer, at the
5 software-based High Level Trigger (HLT) also the information from the Inner Detector
6 (ID) is available, so that the first online reconstruction of tracks, tracking, happens at
7 this stage. Within the HLT the process that requires the most CPU resources is indeed
8 tracking, which consists of two sequential steps: Fast Tracking and Precision Tracking.

9 The Fast Track Finder (FTF) initially finds sets of three spacepoints which may be
10 compatible with the passing of a charged track and then identifies track candidates ex-
11 tending the triplets to the remaining layers. The Precision Tracking stage then selects
12 the highest quality tracks and applies a fit to them.

13 In Run 2 data, it has been observed that the time required for the TFT algorithms
14 follows a power law dependency on the average pile-up level, quantified by $\langle\mu\rangle$, which is
15 defined as the mean number of pp interactions associated with the collisions of proton
16 bunches. This power law behaviour, which can be observed in the right-hand plot of
17 figure 1, is due to the combinatorial nature of the FTF algorithms, so that increasing
18 the space-point density, with higher $\langle\mu\rangle$, a way higher number of combinations will arise,
19 hence slowing down the process.

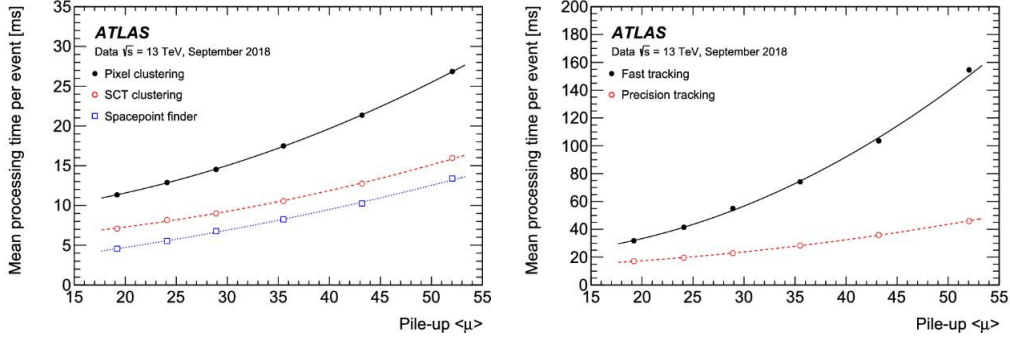


Fig. 1.: The plots show the mean of the total processing time per event, as a function of $\langle\mu\rangle$, for the various tracking algorithms. We notice that the data preparation processes (left) take considerably less time compared to the FTF (right), which shows a power-law behaviour. [1]

2. – A new approach to assist tracking

While the average pile-up obtained in 2018 during Run 2 was $\langle\mu\rangle = 36.1$, with the upgrade to HL-LHC values between 140 and 200 are expected, so that the current CPU farm handling the algorithms running at the HLT, would not be able to support the computing cost. Therefore, a rethinking of the trigger strategies is underway, which could also involve the usage of heterogeneous hardware, including GPUs and/or FPGAs.

In this context, this work proposes a Machine Learning (ML) algorithm which would receive the space points obtained from the Inner Detector (ID) with the goal of filtering out the points due to pile-up tracks from the region of interest (RoI), thereby reducing the combinatorial complexity within Fast Tracking.

The architecture chosen for this purpose and studied in this work is a Convolutional Neural Network (CNN), since these models can be easily accelerated on FPGA boards through the usage of commercial software. Furthermore, since the algorithm will receive image inputs with fixed dimensions, we expect that the time to filter out pile-up hits won't depend strongly on the hit density of the event.

3. – Toy-model event generator

While the final goal is to apply the algorithm to ATLAS events, we start by training and testing the algorithm on synthetic events generated through a simplified model, in order to test the validity of this idea. The hits included in the generated events are divided into two categories:

- Signal: obtained by intersecting charged tracks with the ID layers;
- Background: noise hits randomly generated without considering the correlation between hits from different layers.

In order to simulate the RoI structure the hits are generated within an angular cone defined by $\Delta R \leq 0.8$, but are then projected on the xy plane, in order to obtain a 2D representation that could then be easily supplied to the CNN architecture, which is designed to handle image inputs.

47 To approach a more realistic description, we consider an ATLAS Monte Carlo (MC)
 48 simulation of $t\bar{t}$ jets and calculate the hit density per layer and as a function of the pile-
 49 up. We use these parameters to generate the synthetic events, so that they can mimic the
 50 hit density corresponding to different values of $\langle\mu\rangle$ for MC generated events, as shown in
 51 figure 2.

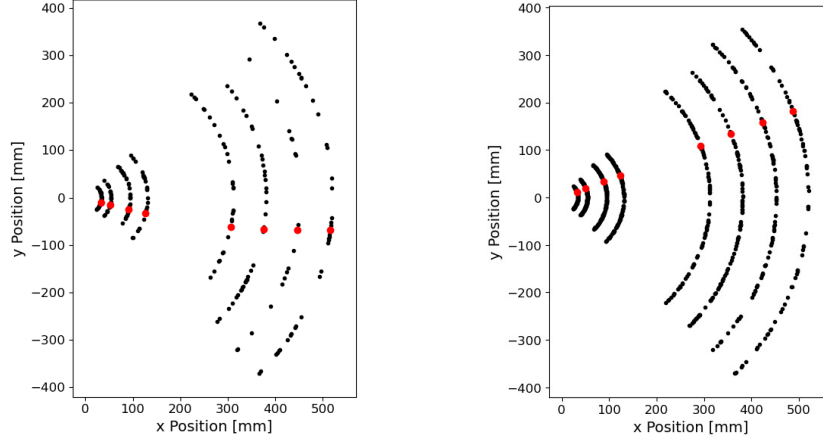


Fig. 2.: Display of two event examples from the toy model generator, with signal hits in red and background hits in black. The event on the right is generated using the hit density values relative to $\langle\mu\rangle=10$, while for the event on the left $\langle\mu\rangle=40$.

52 The change of representation from the (x, y) to the $(\phi, Layer)$ coordinates allows to
 53 better distinguish the hits as separate pixels and is also more suited for an image input,
 54 as can be seen in figure 3. In order to partition the ϕ axis, a binning is introduced: in
 55 this study the bin-width is set to 0.02 radians, which corresponds to $68\ \mu\text{m}$ in the first
 56 layer and about $1\ \mu\text{m}$ in the eighth layer.

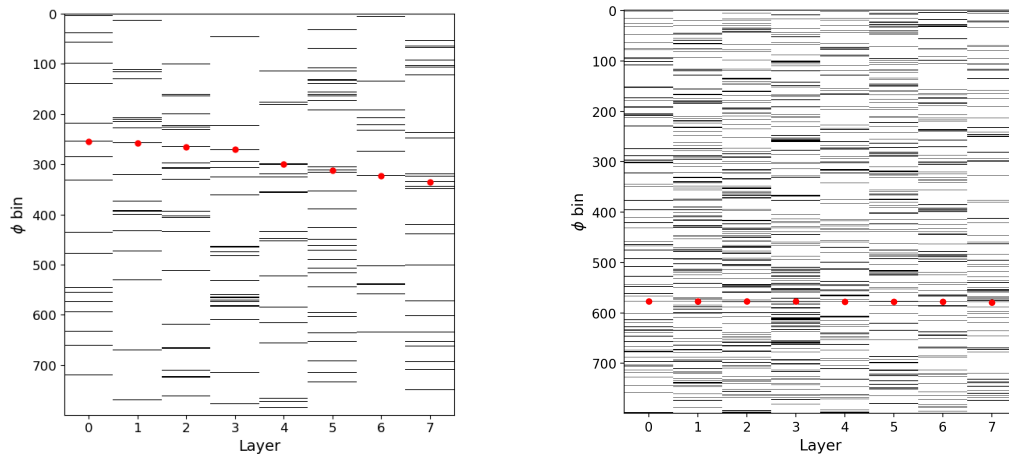


Fig. 3.: The same events from figure 2 are now displayed in the $(\phi, Layer)$ coordinates.

The event on the left of figure 3 shows a low p_T signal track, where the signal hits change their ϕ values across layers. In contrast, high p_T signal tracks maintain aligned ϕ values across layers, forming a more recognizable pattern. Consequently, the algorithm is expected to perform better for high p_T signal tracks.

4. – The CNN Filtering Algorithm

After describing the event generator, we now characterize the CNN algorithm. The network receives input matrices where bin contents are set to 1 if the bin is associated with at least one hit, either signal or background, while the remaining bins have a value of 0. The goal of the CNN is to output a matrix of the same dimensions as the input, with bin contents equal to 1 for the pixels associated with signal hits, and instead 0 for pixels associated with background hits.

This task is analogous to the task of Denoising Autoencoders [3] [4], which is to reconstruct a clean version of the noisy input data by learning to distinguish between signal and noise. For this reason the CNN filtering algorithm draws inspiration from the architecture of Denoising Autoencoders, which is characterized by input compression in the Encoder phase and the return to the input size in the Decoder phase.

The architecture of the filtering algorithm, as shown in figure 4, alternates the Convolutional layer with the Max Pooling layer [5] [6] in the encoder phase, and with the Upsampling layer [6] [7] in the decoder phase.

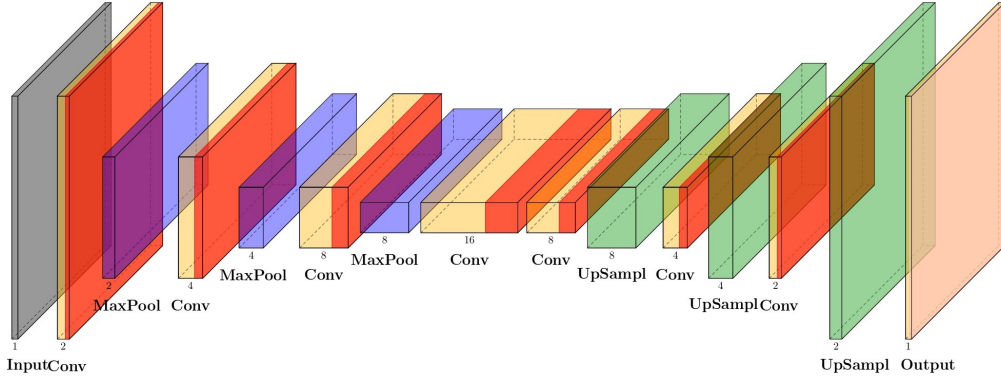


Fig. 4.: Schematic representation of the architecture of the CNN filtering algorithm. The red color associated with the Convolutional layers represents the ReLU function.

The described model is quite simple, allowing much room for improvement. The model is also very lightweight, containing only 12 thousand parameters, which would make it particularly well-suited for fast inference. For comparison ResNet-50 [8], which is a CNN architecture widely used for image recognition tasks, has 25.6 million parameters.

The loss used to train the model is a Mean Squared Error (MSE) loss:

$$(1) \quad \mathcal{L} = \sum_i (y_i - \hat{y}_i)^2,$$

which is computed between the output values \hat{y} and the target values y of the bins containing either signal or background hits, which are indexed by i .

83 The training dataset is composed of 1 million events with a single signal track of
 84 $p_T \in [0.5 \text{ GeV}, 50 \text{ GeV}]$, and the training occurs for 50 epochs using the Adam optimizer.
 85 The trained network is then tested on an independently generated sample, and the
 86 distribution of the output of the bins is shown in figure 5. This plot shows that the
 87 network has learned the filtering task, since most of the background hits have value 0 in
 88 output the majority of the signal hits receive values close to 1.

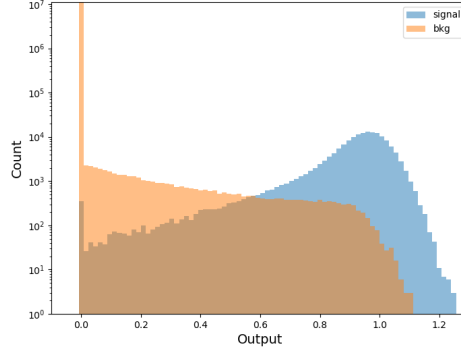


Fig. 5.: Distribution of the output obtained from the filtering CNN for both signal and background hits. [6]

89 It's now possible to use the bins output value as the discriminant between the signal
 90 and background categories, so that the bins with output below a defined y_{cut} are associ-
 91 ated to noise hits, while the bins with $\hat{y} \geq y_{cut}$ are classified as signal.

92 We now define the efficiency of the filtering algorithm as the ratio between the signal
 93 hits correctly classified by the algorithm and the total number of signal hits in the event,
 94 which is 8. We want to investigate how the efficiency is dependent on the p_T value of
 95 the signal track, and to do so we start by defining the Working Points (WPs) for this
 96 evaluation. To define the WPs we test the model on a sample with signal tracks of
 97 $p_T \in [40 \text{ GeV}, 50 \text{ GeV}]$, for which we expect high efficiency values, and look at the y_{cut}
 98 values that correspond to an average efficiency of 90%, 95% and 98%.

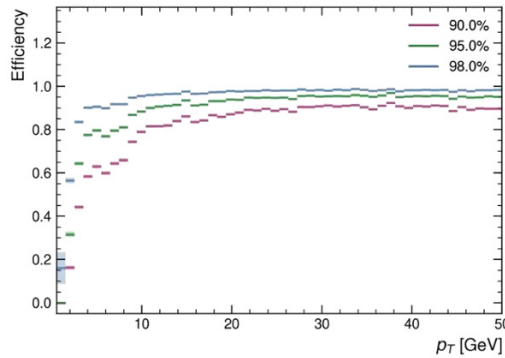


Fig. 6.: Efficiency as a function of p_T of the signal track in the event, computed for the 90%, 95% and 98% WPs. These WPs correspond to the mentioned efficiencies when integrated in between 40 and 50 GeV. [6]

Using the values of y_{cut} obtained, we now use this values for the classification of the $p_T \in [0.5 \text{ GeV}, 50 \text{ GeV}]$ sample, and then look at how the efficiency depends on p_T .

We have introduced the 40-50 GeV sample so that the WP value would correspond to the plateau value of the efficiency at high p_T , which would not have happened if we have used the 0.5-50 GeV sample instead.

The behaviour of the efficiency as a function of p_T is displayed in figure 6, where it's clear that the efficiency worsens for lower values of p_T . Nevertheless, in the case of the 98% WP, the algorithm keeps an efficiency above 90% down to 4 GeV signal tracks.

5. – Conclusions

This work presented a novel ML approach using CNNs to classify hits in a simulated tracking detector, distinguishing charged particle signals from noise. This method aims to address the computational challenges of track reconstruction at the trigger level for the future High-Luminosity LHC program. The CNN filtering algorithm shows promising potential, due to the simple model used and the low number of parameters. Future developments include a more realistic event generator, where the background hits will actually be due to pile-up tracks, and the application of the algorithm to ATLAS events.

REFERENCES

- [1] ATLAS COLLABORATION *2008 JINST 3 S08003*, 10.1088/1748-0221/3/08/S08003
- [2] ATLAS COLLABORATION *The ATLAS Inner Detector Trigger performance in pp collisions at 13 TeV during LHC Run 2*, *Eur. Phys. J. C* **82**, **206** (2022)
- [3] G. E. HINTON, R. R. SALAKHUTDINOV *Reducing the Dimensionality of Data with Neural Networks*, *Science* **313**, **504-507** (2006) DOI:10.1126/science.1127647
- [4] VINCENT, PASCAL & LAROCHELLE, HUGO & BENGIO, Y. & MANZAGOL, PIERRE-ANTOINE. *Extracting and composing robust features with denoising autoencoders.*, *Proceedings of the 25th International Conference on Machine Learning.*, **1096-1103** (2008) DOI:10.1145/1390156.1390294.
- [5] LECUN, YANN & BOTTOU, LEON & BENGIO, Y. & HAFFNER, PATRICK. *Gradient-Based Learning Applied to Document Recognition.*, *Proceedings of the IEEE.* **86.**, **2278 - 2324** (1998) DOI:10.1109/5.726791.
- [6] ZAIO A. *A novel approach for real-time identification of hadronic final states at the High-Luminosity LHC*, <https://cds.cern.ch/record/2903258>
- [7] J. LONG, E. SHELHAMER AND T. DARRELL. *Fully convolutional networks for semantic segmentation.*, *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*, **3431-3440** (2015) DOI:10.1109/CVPR.2015.7298965.
- [8] K. HE, X. ZHANG, S. REN AND J. SUN *Deep Residual Learning for Image Recognition*, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*, **770-778** (2016) DOI:10.1109/CVPR.2016.90.