IL NUOVO CIMENTO **43 C** (2020) 61
DOI 10.1393/ncc/i2020-20061-0

COLLOQUIA: IFAE 2019

Ultra-fast deep learning algorithms on FPGA for the phase-II level-0 trigger of the ATLAS experiment

L. SABETTA

Dipartimento di Fisica, Sapienza Universitá di Roma and INFN, Sezione di Roma - Rome, Italy

received 8 June 2020

Summary. — The LHC accelerator will face, during the following years, a complete upgrade with the main purpose of rising up the instantaneous luminosity by a factor of almost five. Though this will permit to collect an incredible amount of data, the complexity of each event will greatly intensifies going from an average number of interactions per bunch crossing of 40 to an average of 200. To cope with this problem and be able to handle this large amount of information, both the detectors and the trigger algorithms of the ATLAS experiment will be updated. A machine learning approach for the level-0 trigger algorithm is presented.

1. – Introduction

The discovery power of a high-energy physics experiment is directly related to the amount of data collected. For this reason, the Large Hadron Collider (LHC) [1] will undergo through a full upgrade process from 2024 to 2026, which will increase the instantaneous luminosity from the current peak value of $2 \cdot 10^{34}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$ to the expected value of $7.5 \cdot 10^{34}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$. To be able to handle the incredible amount of data produced during the High-Luminosity (HL) phase of LHC [2], the detectors of the ATLAS experiment [3] will be upgraded as well: the central changes related to the subject of this work will be the addition of a Resistive Plate Chamber (RPC) station in the barrel region of the Muon Spectrometer (which will be the innermost, BI) and the operation of the trigger in a Field Programmable Gate Array (FPGA) off detector. Figure 1 shows the RPC detectors of the MS.

2. – Standard trigger algorithm

The Standard Trigger Algorithm (STA), which is thought to be used for HL-LHC, is a direct extension of what has been used before the upgrade, with the addition of the BI station. It operates as a pattern finding algorithm: chosen from which of the detector layer to start, for each hit found, a coincidence window is opened toward the

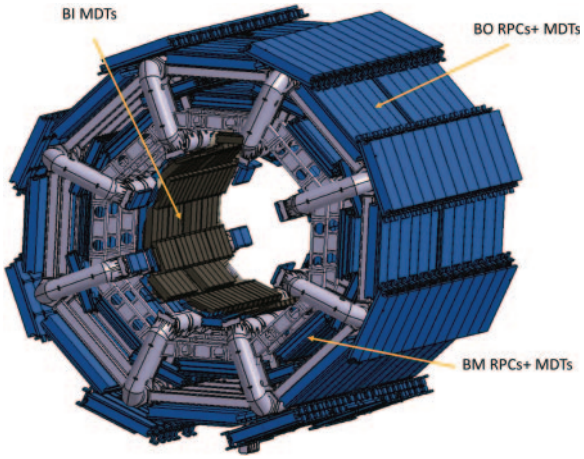

Fig. 1. – Schematic representation of the 3 RPC stations plus the new innermost one, placed in correspondence of the BI MDTs.

adjacent layer. The dimension of the window is inversely proportional to the transverse momentum (p_T) threshold of the trigger so that if the muon p_T is not high enough, the magnetic field will curve the particle outside the coincidence window of the next layer. If a new hit is found, the process is recursively repeated until all the layer are analyzed.

It is then possible to require a minimum number of hits to consider the event triggered, obtaining different trigger configurations. In the following, the configuration which requires at least three coincidences out of the four RPC stations will be used as a performance comparison. Figure 2 gives a schematic representation of the functioning of the STA.

This algorithm is fast and reliable and, being tested for long, well known. Indeed it presents some limitations, being:

- the geometrical acceptance of the RPCs puts an upper limit on the maximum efficiency obtainable;
- the coincidence windows need to be tuned and are susceptible to small variations;
- the STA only gives as output an interval in which the muon p_T is expected to fall in;
- the STA by construction expects particles to come from the primary vertex, limiting the possibility to trigger exotic events.

3. – Machine Learning approach

The limitations of the STA have been solved by applying a Machine Learning (ML) approach to the problem: using a Convolutional Neural Network (CNN) it is possible to allow a treatment of the RPC strips by the trigger system similarly to pixels of a picture. CNNs performances on image recognition tasks are indeed astonishing and well known and studied. As shown in fig. 3, detector strips are therefore arranged in image-like objects, to be fed to the CNN as training inputs. Each bin of the y-axis corresponds to

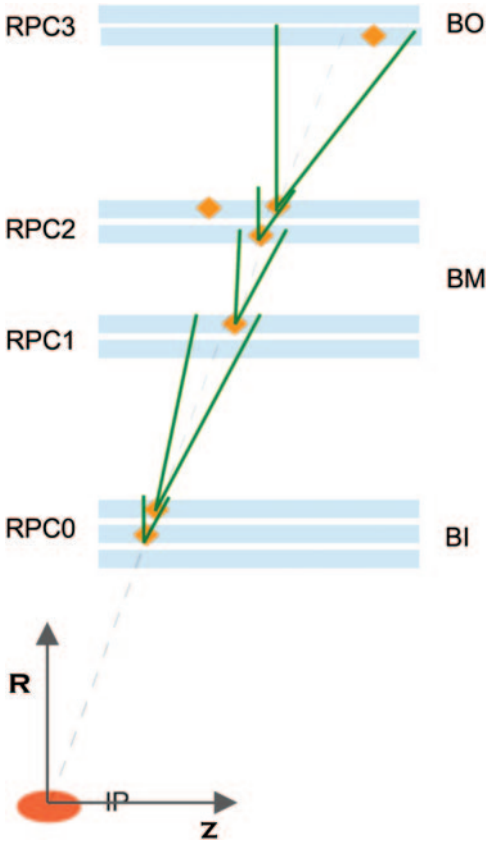

Fig. 2. – Schematic representation of the functioning of the standard trigger algorithm.

a detector layer (3 detector layers for inner station, 4 for the middle and 2 for the outer station). The x-axis maps the η coordinates of each physical RPC strip: for the i-th strip $\eta^i_{bin} = 384\frac{\eta^i - \eta^{min}}{\eta^{max} - \eta^{min}}$, where η^{max} and η^{min} are, respectively, the maximum ($\eta^{max} = 0.95$) and the minimum ($\eta^{min} = 0.07$) η values for the barrel RPC strips chosen to prevent muons to fall outside any layer of a specified sector; 384 is a realistic number of strips per layer. This particular choice has been taken in order to evaluate ML algorithm performances, without any geometrical acceptance effect. Events used in the training phase of the CNN contain:

- just cavern background;
- one muon plus background;
- two muons plus background;
- three muons plus background.

Events with more than one muon are built superimposing one-muon images with no background, which is then included. Cavern background has been evaluated from minimum bias events at HL-LHC pile-up conditions, and then simulated. A total of one

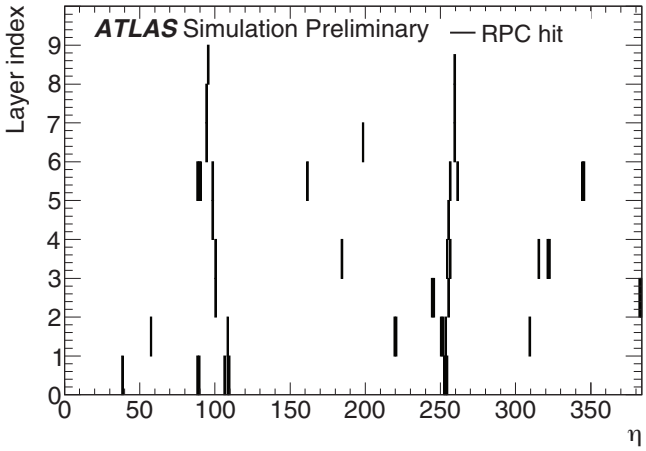

Fig. 3. – Example of one of the images to be fed to the neural network which contains two muons with a p_T equal to 17 and 13 GeV plus background.

million images with muon in the range 3–20 GeV p_T has been used, splitted between training, validation and testing.

The CNN gives as output for each input a 5-component vector:

$$(p_T^{lead},\ \eta^{lead},\ p_T^{sub\text{-}lead},\ eta^{sub\text{-}lead},\ n^{muons}) \tag{1}$$

where lead stands for leading (*i.e.*, the muon with the highest p_T) and n^{muons} represents the number of muons in an image.

The kind of CNN chosen is a ternary Convolutional Neural Network (tCNN) [4]: each of the weights of the network can assume only three values, -1, 0, 1. In this way they can be represented using just two bits per weight instead of 32 as it would have been for floating point weights. This choice becomes fundamental when it comes to implementation: since the FPGA has a limited memory, reducing the occupancy permits to obtain a greater parallelization and consequently greater performances in term of latency. In fig. 4 the architecture of the tCNN is reported.

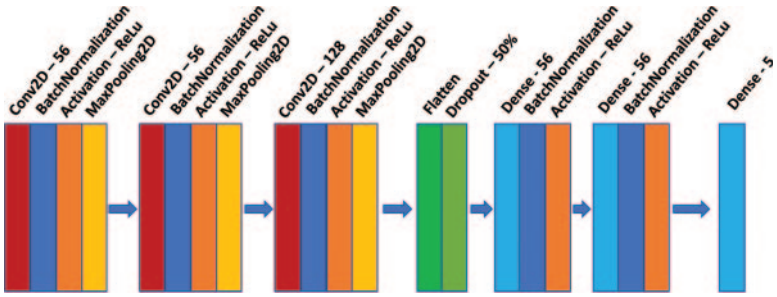

Fig. 4. – Depiction of the architecture that has been adopted. The numbers beside "Conv 2D" represent the number of filters. The numbers beside "Dense" represent the number of neurons of that layer.

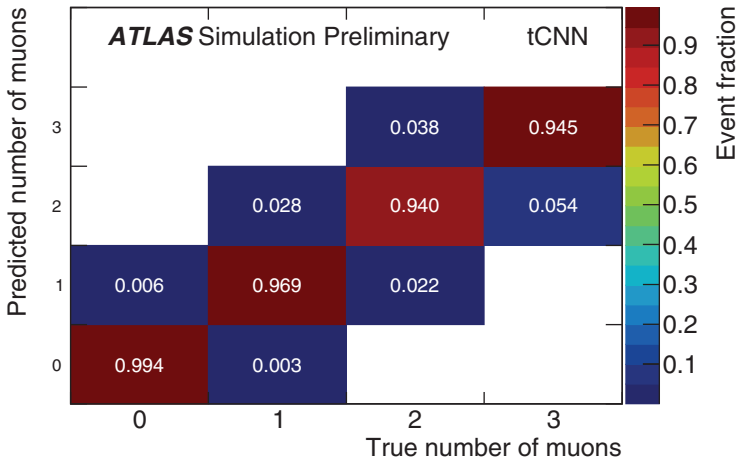

Fig. 5. – Confusion matrix of predicted number of muons *vs.* offline reconstructed number of muons.

In fig. 5 the performance of the network in terms of the number of muons identified by the tCNN *vs.* the number of muon reconstructed offline is reported. Each column is normalized to unity. No trigger threshold is applied for the realization of this plot: by applying a threshold the off-diagonal numbers are reduced even more. In particular, in the case where 0 muons are reconstructed in the detector and 1 muon is reconstructed by the network, requiring a minimum p_T of 10 GeV, the number decreases from 0.6% to 0.01%.

4. – FPGA implementation

For the implementation into the FPGA the algorithm needs to be written in VHDL language. For this purpose, a tool produced by Xilinx [5] has been used, *i.e.*, "Vivado HLS", which takes as input a C/C++ code and turns it into a VHDL code. Therefore the first step after the performance evaluation of the network has been the translation of the algorithm from the original Python form to a C++ code. During this phase various choices have been taken in order to improve the performances in terms of latency: the total time elapsed per event must indeed stay under the timing requirement of the ATLAS experiment, *i.e.*, $\sim$1.5 μs. Apart from specific parallelization and memory usage directives, an important decision has been to modify the architecture into a reduced one which takes as input a smaller section of the event at a time and operates scanning the whole image section by section. Passing a smaller input to the tCNN widely reduces the total number of multiplications and therefore, memory occupancy.

In fig. 6 different efficiency curves are shown. In cyan the efficiency of the STA in the configuration "3 out of 4" as a function of p_T is shown. In blue the efficiency obtained with the tCNN which takes as input the whole image. On the left, in red, the efficiency obtained with a CNN with floating point weights. On the right, in magenta, the efficiency obtained using the tCNN which takes as input just a section of an image at a time but with an even simpler structure than the one showed in fig. 4. The network with the last configuration has effectively been implemented inside an FPGA within the timing requirement ($\sim$1 μs). All the curves are realized taking single-muon images without

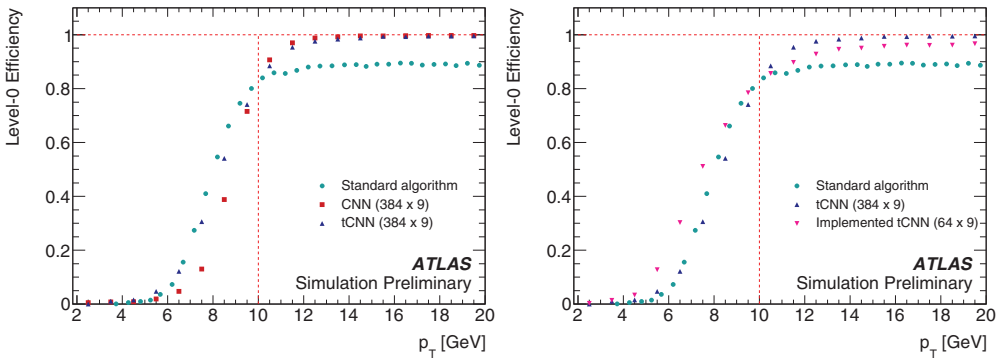

Fig. 6. – Trigger efficiency of different algorithms as a function of the reconstructed p_T.

background as input, and are tuned so that the efficiency reached at nominal threshold (10 GeV) is the same (82%) to ease the comparison. As can be easily seen from the plot on the left, the CNN always overperforms the STA (lower efficiency under threshold and higher efficiency above threshold) with a small decrease in performance turning to tCNN. The plot on the left shows that even if some optimization is still required to reach the performance of the bigger tCNN, the "implemented tCNN" still obtains comparable results with respect to the STA.

5. – Conclusions

Different ML alternatives to the STA have been presented for the phase-2 level-0 trigger of the ATLAS detector. In particular CNNs are capable of overcome the problems presented by the STA, with comparable or even greater performance in terms of efficiency:

- tCNNs are not limited by geometrical acceptance;
- given the generic structure of the output, a tCNN is much more flexible than the STA;
- the tCNN performs regression on the p_T, and does not just give an interval;
- tCNN does not expect by construction particles to come from the primary vertex opening the possibility to use dedicated trigger for exotic events.

More optimization strategies are planned to be investigated in the future in order to be able to synthesize the best performing tCNN into the FPGA.

REFERENCES

[1] Evans Lyndon and Bryant Philip, *JINST*, **3** (2008) S08001.
[2] ATLAS Collaboration, *Technical Design Report for the Phase-II Upgrade of the ATLAS Muon Spectrometer*, Technical Report, CERN-LHCC-2017-017, ATLAS-TDR-026 (CERN) 2017.
[3] Aad G. *et al.*, *JINST*, **3** (2008) S08003.
[4] Li Fengfu and Liu Bin, *Ternary Weight Networks, CoRR*, abs/1605.04711 (2016).
[5] *Vivado Design Suite User Guide - High-Level Synthesis* (Xilinx) 2012.