Artificial Intelligence Research and Development
I. Sanz et al. (Eds.)

doi:10.3233/FAIA230694

Reconstruction of the LHCb Calorimeter Using Machine Learning: Lessons Learned

Núria VALLS CANUDAS [a,1], Xavier VILASIS-CARDONA [a], Míriam CALVO GÓMEZ [a] and Elisabet GOLOBARDES RIBÉ [a]

[a] *Smart Society Research Group, Engineering Department, La Salle-Universitat Ramon Llull, Sant Joan de La Salle 42, 08022 Barcelona, Spain*

ORCiD ID: Núria Valls Canudas https://orcid.org/0000-0001-8748-8448, Xavier Vilasis-Cardona https://orcid.org/0000-0002-1915-9543, Míriam Calvo Gómez https://orcid.org/0000-0001-5588-1448, Elisabet Golobardes Ribé https://orcid.org/0000-0001-8080-0769

Abstract. In particle physics experiments, calorimeters are in charge of measuring the energy of incoming particles. In order to correctly estimate and evaluate the energy and other properties of these particles, a process, called reconstruction, is required. Because of the amount of collisions and the data-flow, reconstruction algorithms need to be time savvy. The nature of the problem seems appropriate for deep neural networks, yet the approach shows constraints. This paper presents the application to the calorimeter of LHCb in the first upgrade phase under the so-called Real Time Analysis framework, which is in charge of processing 30 MHz of data in real time, with its pros and cons.

Keywords. High Energy Physics, Calorimeter, Deep Learning, Optimization, Model Inference

1. The calorimeter reconstruction challenge

In the field of high energy physics (HEP) there are many computational challenges regarding data. As an example, the LHCb experiment at CERN currently processes 30 MHz of non-empty proton-proton collisions, which generate a data flow of 5 TB/s produced by the eight sub-detectors readout [2]. Far beyond the limits of storage technology, this rate is filtered by a factor 400 by the called *trigger system* [1] before its storage. This trigger relays on a full reconstruction of the collision events in order to make specific selections of physics signatures to decide which events are stored for further analysis. There are two stages in the trigger system: the High Level Trigger 1 (HLT1) [4], which processes the full detector readout in a GPU framework and performs a partial reconstruction of events and primarily selections; and the High Level Trigger 2 (HLT2) which makes a full reconstruction of events using a quasi-real-time alignment and calibration of the detector together with the order of a thousand specific selection algorithms. HLT2 has a throughput of 10 GB/s of reconstructed events and is executed in a CPU framework.

[1] Corresponding Author: Núria Valls Canudas, nuria.valls@salle.url.edu

With independence of the trigger stage, the reconstruction of events consists on a series of specific algorithms that process the data generated by each of the LHCb sub-detectors. By analysing the interaction of particles with the detector, the particles produced in the collision are identified.

One of the eight sub-detectors in LHCb is the electromagnetic calorimeter (ECAL) [10]. It is designed to measure the energy and position of particles as they interact with the detector material with high precision and is the only detector capable to identify neutral particles. It has a rectangular shape of 7.8×6.3 m and is placed perpendicular to the accelerator beam pipe. The ECAL structure is segmented into individual square-shaped modules that perform the energy measurement. Each module has a variable number of readout cells depending on the position, which conform three rectangular regions with increasing granularity around the central beam pipe. The output data obtained from the calorimeter are the values from the 6016 readout cells concerning the accumulated energy deposited by incident particles in a single collision. The energy deposits from a single collision are values from 0 to 10.240 called digits.

Regarding the reconstruction process, it consists on clustering together the energy deposits that belong to the same incident particle, which are then called *clusters*. Although the module shape is designed to contain the full energy of an incident particle, they may not impact at the center of a module. Therefore, the energy shower deposited by a particle is reconstructed as a 3×3 cell group. Given the current LHCb running conditions, ECAL has a high occupancy with the order of 250 clusters per event. This implies that many clusters are overlapping in the inner-most region, which adds more complexity to the reconstruction process.

Entering in the detail of a generic reconstruction for ECAL, it can be segmented into three processes. The first step is to identify the seeds of clusters, defined as the most energetic deposit in a local cell matrix of 3×3. Only deposits of more than 50 MeV are considered to be a seed as ECAL is focused in the reconstruction of photons and electrons that deposit all of its energy in the calorimeter cells. This also helps to filter out the background noise. The second step in the reconstruction chain is to tag the rest of the digits from a single collision to the closest seed. If a digit is in between two seeds, it is tagged to both of them and later processed as an overlapping cell. The final step consists in accumulating the energy of all the digits tagged to a seed and account it as a reconstructed cluster. In case of an overlapping cell, its energy is split according to the energy of the overlapping clusters and each part is linked to the respective cluster.

2. Evolution of methodologies

The process of reconstructing calorimeter data can be viewed as a clustering problem, where the primary objective is to group energy deposits from particles according to specific rules. Traditional unsupervised clustering algorithms use extensive recursive functions with distance or density metrics for creating clusters [6]. However, the approach taken for calorimeter data reconstruction in LHCb stands apart from classical clustering algorithms. This distinction arises due to the strong physics and execution time requirements.

Focusing on the field of calorimetry in High Energy Physics, the Cellular Automaton has been a benchmark solution for many years [3] in LHCb. A more recent optimization

of the latter, an algorithm named Graph Clustering [14] is the current solution for the ECAL reconstruction in HLT2. It is based in the use of graph data structures to store the energy deposits from an event, such that the related digits are closely linked in the graph. This allows for a fast and efficient processing of the overlapping clusters and a flexible representation of data. Although Graph Clustering clearly outperforms the previously used method in terms of execution time, it still has a quadratic complexity curve that grows with the number of energy deposits. This points out the vision of future LHCb upgrades in which the occupancy of the detector is expected to increase by a factor of 10 [8].

To keep up with such rates, there are two main approaches that can flatten the complexity curve. The first one is to take advantage of parallel architectures such as GPUs and FPGAs, which requires a non-trivial translation of the algorithms to take advantage of small local operations that are highly parallelizable. In this direction, some efforts are currently being made in order to adapt the logic of the Graph Clustering algorithm into the preliminary ECAL reconstruction algorithm in the HLT1 GPU framework [15].

On the other hand, the second approach to reduce the reconstruction complexity is through the use of deep learning models. If the ECAL readout data from a collision would be modeled as an image, it would have 6016 pixels with values from 0 to 10.240 from which the reconstructed clusters energy and positions need to be extracted.

It has long been demonstrated that artificial intelligence is well suited for many different HEP challenges [11,9]. Specifically, deep learning models have proven to solve many complex issues at very high speeds, only at the cost of increasing the time and complexity of the training. However, complex models tend to explode in number of parameters, which end up causing the inference to be slow. There is a current trend in HEP to make an efficient use of deep learning models by optimizing the networks itself. This optimization is achieved when the needs of the problem to solve are well understood and the network is modeled and trained accordingly.

On this same line, there is a proposed solution for the ECAL reconstruction that uses a sequence of two convolutional neural networks [13]. The main aspect of such approach is that segmenting the reconstruction process into steps allows to train neural networks on the rules that solve a general formulation of the problem. Furthermore, the understanding of the local nature of the problem leads to a simplification of the data-set, where thousands of training samples are extracted from only 2000 full ECAL simulations. Results regarding the inference time using simulation samples of LHCb collisions show a nearly constant behavior towards the number of digits in a collision compared to a version of the benchmark Cellular Automaton algorithm, as shown in Figure 1.

3. Limitations and constraints

In the latest years, there has been an increasing interest in machine learning and deep learning inference engines as a tool for fast deployment of AI models into production. One of the standards used in HEP is the TMVA tool for performing multivariate analysis in the ROOT framework [7]. It has long been used in LHCb for offline analysis of large data samples using mainly boosted decision trees (BDT) and multi layered perceptrons (MLP). But precisely because this tool is thought to be used in offline analysis, it priori-

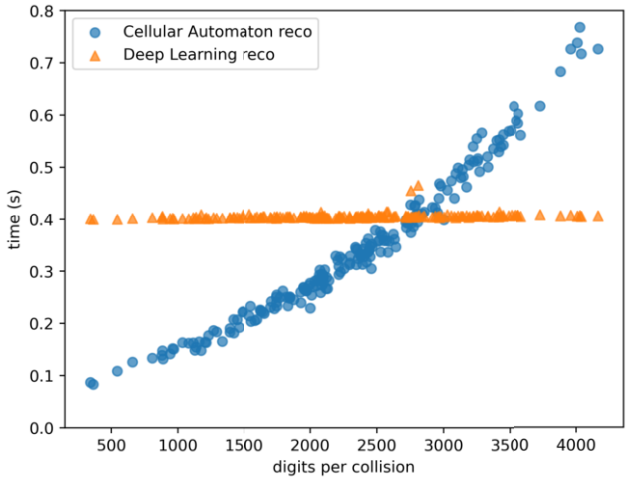

Figure 1. Scatter plot of the mean computational time over the number of digits per collision from LHCb simulations. Comparing a Python version of the LHCb benchmark Cellular Automaton and the simplified deep learning implementation [13].

tizes easy usability rather than inference performance. As an example, a fast neural network based algorithm was proposed in 2017 for LHCb [5]. However the use of TMVA tools was not sufficient to cope with the HLT2 throughput requirements and it required by-hand modifications of the code to allow auto-vectorization and further optimizations of the implementation.

Since this is not scalable and hard to maintain, other tools have started to come into play. Focusing on the model deployment, ONNX is an open format built to represent machine learning models. It defines the building blocks of ML and DL models as common operators and creates a common file format that allows to use the AI models in different frameworks. On the same line, TensorRT is a software development key from NVIDIA that provides high-performance deep learning inference for CUDA environments. It also allows to read and use ONNX files.

Taking advantage of the NVIDIA tools for CUDA, recent studies have used TensorRT to test the inference of two benchmark dense neural networks in the HLT1 GPU platform in LHCb [12]. The two networks tested are both MLP architectures using 17 input features from the LHCb tracking detectors. Looking at the overall results, the kernel overhead is the main bottleneck for throughput but large batch sizes minimize the throughput decrease. To give more detail, the first model tested is a dense neural network with two hidden layers. With the order of 1000 parameters, the HLT1 reconstruction sequence shows almost no throughput reduction using one instance of the network. However, compared to the second, larger approach, with six hidden layers of up to 128 neurons each, the throughput shows a decrease of almost 5%.

We can extrapolate those numbers to the simplified deep learning approach for the ECAL reconstruction. Recalling the network architectures of the approach, it consists of a first convolutional neural network with 1272 parameters for each of the three ECAL regions. The input data is represented as images of sizes 64×52, 64×40 and 48×36

pixels for the Outer, Middle and Inner regions respectively. As a second sequential step, the data from the first networks is processed in windows of 7×7 cells to isolate the relevant information for a single cluster. Then, this window is convoluted with a MLP kernel of size 5×5, which has $108,993$ parameters. Further details on the network chain can be found in the original paper [13]. Overall, the overhead of the data processing and the MLP network inference can be approximated to have the same cost as the six hidden layer model tested in HLT1. To have a broad estimation of the whole impact of the ECAL reconstruction approach, we need to add the inference cost of the first CNN. Although it is negligible for one instance, the approach has one instance per region, which has an impact of almost 6% to the total throughput. Therefore, with a broad estimation, we can say that the DL approach for ECAL reconstruction would imply a throughput reduction of 11% of the whole HLT1 sequence when executed inside the GPU framework of LHCb.

This high inference impact is further enhanced when compared with the cost of the current calorimeter reconstruction in HLT1, which uses a set of CUDA algorithms to find the cluster seeds and build 3×3 clusters around them. This simplified algorithm that does not take into account the overlap between clusters represents a 4% of the total HLT1 reconstruction throughput.

4. Discussion and conclusions

All the algorithms currently implemented for HLT1 and HLT2 have a strong dependency in the number of digits in a collision. Even tho the parallelization of CUDA algorithms can help to mitigate this effect, the future prospects for the LHC experiments include a major upgrade in which the objective is to increase the number of particles generated in every collision by a factor of 10. In such a scenario, a deep learning model with nearly constant inference time, regardless of data complexity, becomes highly valuable.

However, although the many application of AI models in HEP have demonstrated to be very effective in data analysis and reconstruction, the inference of such models to the real experiments frameworks is a clear bottleneck.

Through the study of the LHCb calorimeter reconstruction, we have seen that the standard HEP tools for the inference of models are not scalable and require an expert knowledge in advanced code optimization which sets a huge barrier for deploying or testing the models.

On the other hand, newer tools are starting to be mature enough to allow a generalized format for inferencing AI models that provide fast inferences in an optimized environment. However, the ECAL reconstruction process requires a set of complex operations that are non trivial for an AI application. Even with an optimized and simplified network architecture, the resulting algorithm is expected to have a non-negligible decreasing effect on the throughput even with parallel architectures.

The first outcome we want to raise is that deep learning models will only be suited for HEP trigger-like applications if they are small, simplified and well optimized. This can only be achieved when the insights of the problem are well understood and a network is modeled according to them, instead of expecting a model to learn the general rules of the problem by itself.

As a second message, it is key for the future development of deep learning applications in HEP to push the development of fast and optimized tools for inferencing AI models inside HEP frameworks either with CPUs or GPUs.

Acknowledgement

This work is supported by MICINN under Grant PID2019-106448GB-C32 and by Generalitat de Catalunya under Grant 2021 SGR 01398.

References

[1] Aaij, A.; Benson, S.; De Cian, M.; Dziurda, A; Fitzpatrick, C.; Govorkova, E.; Lupton, O.; Matev, R.; Neubert, S.; Pearce, A.; Schreiner, H.; Stahl, S. and Vesterinen, M. *A comprehensive real-time analysis model at the LHCb experiment*; *Journal of Instrumentation*, **2019**, *14(04)*.

[2] Bediaga, I.; Chanal, H.; Hopchev, P.; Cadeddu, S.; Stoica, S.; Calvo Gomez, M.; T'Jampens, S.; Machikhiliyan, I.V.; Guzik, Z.; Alves, A.A., Jr.; et al. *Framework TDR for the LHCb Upgrade: Technical Design Report*; Technical Report; LHCb-TDR-012; Geneva, CERN 2012.

[3] Breton, V.; Brun, N.; Perret, P. *A Clustering Algorithm for the LHCb Electromagnetic Calorimeter Using a Cellular Automaton*; Technical Report; CERN-LHCb-2001-123; Geneva, CERN 2001.

[4] The LHCb Collaboration. *LHCb Upgrade GPU High Level Trigger Technical Design Report*; Technical report, CERN-LHCC-2020-006, LHCb-TDR-021, CERN, Geneva, 2020.

[5] De Cian, M.; Farry, S.; Seyfert, P. and Stahl, S. *Fast neural-net based fake track rejection in the LHCb reconstruction*; Technical report, CERN-LHCb-PUB-2017-011, LHCb-PUB-2017-011, CERN, Geneva, 2017.

[6] Han, J.; Pei, J. and Kamber, M. *Data mining: concepts and techniques*; *Elsevier*, **2011**

[7] Hoecker, A.; Speckmayer, P.; et al. *TMVA - Toolkit for Multivariate Data Analysis*; *arXiv preprint physics/0703039*, 2009.

[8] The LHCb Collaboration. *Framework TDR for the LHCb Upgrade II - Opportunities in flavour physics, and beyond, in the HL-LHC era*; Technical report, CERN-LHCC-2021-012, LHCB-TDR-0231, CERN, Geneva, 2021.

[9] Mazurek, M. *Deep learning solutions for 2D calorimetric cluster reconstruction at LHCb.*; *4th Inter-experiment Machine Learning Workshop*; Talk; LHCb-TALK-2020-178; 2020.

[10] Omelaenko, O.; Dalpiaz, P.; Guzik, Z.; Spiridenkov, E.; Jarron, P.; Semenov, V.; Ocariz, J.; Khan, A.; Perret, P.; Schneider, O.; et al. *LHCb Calorimeters: Technical Design Report*; Technical Report; LHCb-TDR-002; Geneva, CERN 2000.

[11] Qasim, S.R.; Kieseler, J.; Iiyama, Y.; Pierini, M. *Learning representations of irregular particle-detector geometry with distance-weighted graph networks.*; *The European Physical Journal C*, **2019**, *79(7), 1-11*.

[12] Sclocco, A.; Ali, S.; van den Oord, G.; Cámpora Pérez, D.; de Vries, J.; Aaij, R. and van Veghel, M. *High-throughput machine learning inference with NVIDIA TensorRT*; Presented at *26th International Conference on Computing in High Energy and Nuclear Physics (CHEP23)*, Talk; Norfolk, Virginia, USA; 2023.

[13] Valls Canudas, N.; Calvo Gómez, M.; Golobardes Ribé, E.; Vilasis-Cardona, X. *Use of Deep Learning to Improve the Computational Complexity of Reconstruction Algorithms in High Energy Physics.*; *Applied Sciences*, **2021**, *11(23), 11467*

[14] Valls Canudas, N.; Calvo Gómez, M.; Vilasis-Cardona, X.; Golobardes Ribé, E. *Graph Clustering: a graph-based clustering algorithm for the electromagnetic calorimeter in LHCb.*; *European Physical Journal C*, **2023**, *83, 179*

[15] Valls Canudas, N.; Calvo Gómez, M.; Vilasis-Cardona, X.; Golobardes Ribé, E. *Preliminary performance study of an alternative calorimeter clustering solution for Allen in LHCb*; Presented at *26th International Conference on Computing in High Energy and Nuclear Physics (CHEP23)*, Talk; Norfolk, Virginia, USA; 2023.