Application of distance-weighted graph networks to real-life particle detector output

Yutaro Iiyama **Gianluca Cerminara** **Abhijay Gupta** **Jan Kieseler**
CERN, Geneva, Switzerland

Maurizio Pierini **Marcel Rieger** **Shah Rukh Qasim** **Gerrit Van Onsem**

Kinga Wozniak

Abstract

Real-life applications of recently proposed graph-based neural network architectures are investigated. Sensors of the planned next-generation calorimeter of the CMS detector at the CERN Large Hadron Collider are represented by graph vertices, embedded into high-dimensional latent spaces, where features associated to each vertex are updated according to spatial distances calculated in these spaces. Challenges in applying these algorithms to a real-world detector are described, and concrete applications are laid out.

1 Introduction

Machine learning (ML) has been, and will increasingly be, a key ingredient to all aspects of high-energy physics (HEP) experiments, such as those conducted at the CERN Large Hadron Collider (LHC). HEP data processing routinely involves tasks such as reconstruction of particle trajectories through the detectors (clustering), identification of particle types (classification), and measurement of particle energy and momentum (regression), which are all essentially high-level pattern recognition problems. Traditionally, domain-specific algorithms would perform the basic parts of such tasks, extracting high-level features from the raw input, and ML algorithms such as Boosted Decision Trees are applied to these features to enhance the precision of the final results. However, as a result of both the increasing complexity of the particle detectors and the technological improvements in the domain of deep learning (DL), more holistic, end-to-end approach, where the raw input is directly passed to ML algorithms, is gaining attention.

So far, common methods of raw-input processing with DL in HEP have been to cast the particle detector readout as a two- or three-dimensional pixelated image, on which a standard convolutional neural network (CNN) is applied [1; 2; 3; 4]. While this approach benefits from having highly optimized CNN libraries already available, it may fail to extract the true performance potential of the detector, whose complex three-dimensional geometry usually does not fit readily into a rectangular grid. Moreover, detectors at collider experiments are usually designed so that the occupancy[1] does not exceed 10-20% even in the highest-multiplicity events. This means that the pixel images fed to CNN are sparse, possibly costing more computing resource than necessary for the amount of information being processed.

Learning functions over sparsely and irregularly distributed points is a task that is handled well by graph-based neural networks [5]. Particularly, Qasim et al. [6] recently proposed two novel

[1] Detector occupancy is defined as the ratio of the number of readout channels with particle signals to the total number of readout channels.

Second Workshop on Machine Learning and the Physical Sciences (NeurIPS 2019), Vancouver, Canada.

DL architectures based on graph networks, where detector channels with non-null readout, or hits, are represented as vertices of graphs. The architectures, dubbed GRAVNET and GARNET, translate semantic affinity of the vertices into geometric proximity in high-dimensional latent space, where features associated to each vertex are more strongly affected by objects closer to it (distance-weighted feature update). In GRAVNET, the vertices interact with each other, while in GARNET, the interactions are with central aggregator nodes. Both architectures rely only on commonly available tensor-manipulation operations, such as fully connected layers, that are used in CNNs. However, unlike CNNs, these architectures require no conversion (preprocessing) of detector readout into rectangular-grid images, and are able to maximally utilize the spatial information of irregular detector geometries.

The GRAVNET and GARNET architectures were developed in view of applications for the next-generation endcap calorimeter of the CMS detector, to be installed for the High-Luminosity LHC phase (Phase II) [1]. The High Granularity Calorimeter, or HGCAL, will be a sampling digital calorimeter comprising hexagonal arrays of silicon sensor cells interleaved with absorber layers. In the current design, the smallest sensor cells are $0.52\mathrm{cm}^2$ in area, with the distance between the centers of the nearest-neighbor cells at $0.4\mathrm{cm}$. There will be approximately 6 million readout channels in the HGCAL alone. The HGCAL is also designed to partake in the level-1 (hardware) trigger system of the CMS detector, where potentially interesting proton collision events must be identified from the patterns of energy depositions within $5\mu\mathrm{s}$ latency. Thus, GRAVNET and GARNET must operate over $\mathcal{O}(10^5)$ vertices (considering the detector occupancy) and some version of them must run on field programmable gate arrays (FPGAs).

However, to better focus on the characterizing the network architectures and not the nontrivial HGCAL geometry, the studies reported in Ref. [6] were performed on a simulation of a toy calorimeter, which possesses a layered structure similar to the HGCAL, but is much smaller and is made of non-uniform but rectangular arrays of sensors. Moreover, while the inference times of the GRAVNET and GARNET layers are reported in comparison with other DL architectures, no dedicated attempt was made to optimize the layers for the latency demanded in the level-1 trigger.

In this paper, we report on the progress of the initial studies of applications of GRAVNET and GARNET to the HGCAL. After a summary of the architectures, additional steps to make them applicable to the HGCAL readout are presented. Finally, a list of currently pursued and potential applications is given.

2 The GravNet and GarNet layers

The GRAVNET and GARNET layers both receive as input a $B \times V \times F_{\mathrm{IN}}$ data set, consisting of a batch of B examples, each represented by a set of V detector hits, embedded in the network set through F_{IN} features. For instance, the F_{IN} features could include the Cartesian coordinates of a given sensor, its address (layer number, module number, etc.), the sensor time stamp, the recorded energy, etc. Within the layers, the F_{IN} features are first converted into S-dimensional coordinates in a latent space by a dense[2] layer. Another dense layer reinterprets the same F_{IN} features into F_{LR} features of the vertices in this latent space. The F_{LR} features then undergo a distance-weighted update in the latent space, and are fed into the final dense layer together with the original F_{IN} features, ending up in F_{OUT} features. The output data set thus has a shape $B \times V \times F_{\mathrm{OUT}}$, allowing multiple chained application of the same layers.

The two layers differ in the distance-weighted feature update algorithm. In GRAVNET, each vertex receives the F_{LR} features of the N nearest neighbors in the S-dimensional latent space, weighted (multiplied) by a Gaussian function of the distances to the corresponding neighbor vertices. The weighted features are aggregated over N using the maximum and the mean function, and the result is taken as the updated features of the vertex. On the other hand, in GARNET, S coordinates are interpreted as one-dimensional distances to S aggregator nodes. The F_{LR} features of each vertex are passed to the aggregator nodes with weights given by the negative exponential of the distance to the

[2]Here and in the following, *dense* layer refers to a learnable weight-matrix multiplication and bias vector addition with respect to the last feature dimension, with shared weights over all other dimensions. In this case, the weights and bias are applied to the vertex features F_{IN} and shared over the vertices V. This can also be thought of as a 2D convolution with a 1×1 kernel.

aggregator. At the aggregator, each of F_{LR} features are reduced to their maximum and mean over V. The resulting features are passed back to the vertices, weighted by the same exponential function.

The main advantage of the two architectures comes from the fact that the F_{OUT} output (unlike the F_{IN} input) carries collective information from each vertex and its surroundings, providing a more informative input to downstream processing. Furthermore, the explicit distinction between learned spatial coordinates S and learned features F_{LR} allows a better understanding and control of the network behavior.

3 Supporting developments for real-life applications

3.1 Spherical and quasi-cylindrical coordinates

A spherical coordinate system, given by the distance from the origin r, polar angle θ, and azimuthal angle ϕ, or a quasi-cylindrical coordinate system, given by the distance from the zenith axis ρ, pseudorapidity $\eta = -\ln\tan(\theta/2)$, and azimuthal angle ϕ, are more suitable than a Cartesian coordinate system for describing the positions of the hits in a collider detector. When the original input features F_{IN} are given in terms of such coordinate systems, all mappings of the vertex coordinates into latent spaces should preserve the S^1 topology of the azimuthal coordinate. Coordinate transformation layers using trigonometric functions are newly developed for this purpose.

3.2 Definition of ground-truth

As discussed in Section 4, identification of the species of the particle associated to a cluster of energy deposition is one of the fundamental tasks. However, the concept of particle species is not always clear-cut in a real-world collider environment. A classical example where the particle species becomes ambiguous is an electron that emerges from the interaction point and undergoes a bremsstrahlung, reaching the calorimeter surface accompanied by a near-collinear photon. If the clusters of energy depositions of the electron and the photon do not overlap, it is natural to identify each cluster separately as an electron and a photon. If, on the other hand, the clusters fully overlap, it should be identified as an electron. However, the clusters can also partially overlap, necessitating a threshold for the degree of overlap, beyond which two clusters are associated with one particle, to be set.

While DL models for particle identification cannot be trained without a stable definition of ground-truth, this threshold for the overlap also depends on the ability to resolve overlapping clusters of the actual DL model in use. Therefore, ground-truth definition is considered as an iterative procedure where truth definition itself is tuned together with the model hyperparameters to achieve the ultimate particle identification performance.

3.3 Prediction of unordered sets

The task of cluster reconstruction in the calorimeter can be regarded as a problem of set prediction with unknown cardinality. It is a set prediction because the resulting clusters do not have inherent ordering, and the cardinality is a-priori unknown in an end-to-end reconstruction because only the raw detector readout is provided as the input to the network.

The difficulty of set prediction is in the combinatoric redundancy of the output labeling. Recently, several approaches for solving general set prediction problems have been proposed [7; 8]. However, these algorithms still consider all permutations of output labels at some point in the execution, which becomes a prohibitively expensive operation for reconstructing $\mathcal{O}(10)$ or more clusters. More efficient set prediction algorithms and / or formulations of the cluster reconstruction problem that do not involve set predictions are being investigated.

3.4 Memory footprint reduction

As mentioned in Section 1, the graph networks must process $\mathcal{O}(10^5)$ detector hits. From a resource usage perspective, it is not trivial and may be impossible to fit all hits in one graph network. In particular, the GravNet layer involves a computation of $V \times V$ adjacency matrix, requiring a large amount of RAM.

Two parallel solutions are investigated for this problem. One is to improve the memory efficiency of GRAVNET using variable-length arrays and considering that not all $V \times V$ adjacency values are needed at once. The other solution is to divide the HGCAL into small patches and perform the particle reconstruction in each patch independently, then connecting the pieces back together consistently.

3.5 Firmware implementation

The GARNET layer has been translated to register transfer language (RTL) to be synthesized into FPGA logic with the HLS4ML [9] toolkit. A realistic FPGA implementation of a graph-based network must utilize much less computational resource compared to its counterpart that executes on CPUs or GPUs. Even just for RTL translation, the architecture complexity has to be reduced significantly by e.g. using only the mean aggregation and omitting the appendage of input features to the output array. To further reduce the size of the synthesized gate-level representation, several optimizations, ranging from reducing the floating point precision to using bit shift operations in place of the exponential distance weighting, are being considered.

4 Applications on the CMS HGCAL

The GRAVNET and GARNET layers are designed to be generic pattern recognition engines. With all the supporting developments in the previous section complete, these algorithms can be applied for any combinations of the following tasks.

- **Noise reduction.** Distinguish hits caused by real particles from those due to electronic noise in the sensor or the readout devices.
- **Cluster reconstruction.** Associate hits due to the same incoming particle. In case of overlapping energy depositions from multiple particles, possibly assign energy fractions to individual hits, such that overlapping clusters share parts of the reconstructed energies of the hits.
- **Pileup discrimination.** Pileup interactions are additional particle collisions that occur simultaneously with the collision of interest (hard scattering). The product of pileup interactions are typically low-energy hadrons and photons, which would not penetrate deeply into the HGCAL detector volume. By possibly additionally utilizing the information from the inner tracker of the CMS detector, flag the reconstructed clusters from products of pileup interactions, thereby improving the overall description of the hard-scattering event.
- **Particle species identification.** Given a cluster, associate a species of the particle that made the energy deposition.
- **Energy regression.** Given a cluster, provide a accurate prediction of the energy of the particle that made the energy deposition.

As the installation of the HGCAL is the most significant upgrade of the CMS detector towards the LHC Phase II, there are significant efforts within the CMS collaboration to fully understand the detector behavior and to develop algorithms that optimally exploits its potential. The applications of GRAVNET and GARNET to HGCAL reconstruction will be studied in this context, and the results will be made public through the CMS collaboration in the near future.

5 Conclusion

To effectively process particle detector readout, which is often sparse and is spatially distributed in irregular geometries, graph-based neural network architectures have been proposed as alternatives to the convolutional neural networks. The two architectures, the GRAVNET and GARNET layers, have been demonstrated on toy calorimeter models, and are now being adapted to real-world use in the CMS High Granularity Calorimeter. Progress has been made in understanding features and requirements that do not exist in toy models. The investigations are made in the context of CMS upgrade studies, with the results expected to be reported by the CMS collaboration in public documents.

References

[1] CMS Collaboration, "The Phase-2 Upgrade of the CMS Endcap Calorimeter," Tech. Rep. CERN-LHCC-2017-023. CMS-TDR-019, CERN, Geneva, Nov 2017. Technical Design Report of the endcap calorimeter for the Phase-2 upgrade of the CMS experiment, in view of the HL-LHC run.

[2] F. Carminati *et al.*, "Calorimetry with deep learning: particle classification, energy regression, and simulation for high-energy physics." "Deep Learning for Physical Sciences" workshop at NIPS 2017, 2017.

[3] D. Guest, K. Cranmer, and D. Whiteson, "Deep Learning and its Application to LHC Physics," *Ann. Rev. Nucl. Part. Sci.*, vol. 68, 2018.

[4] L. De Oliveira, B. Nachman, and M. Paganini, "Electromagnetic Showers Beyond Shower Shapes." arXiv:1806.05667, 2018.

[5] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: Going beyond euclidean data," *IEEE Signal Processing Magazine*, vol. 34, p. 18, 2017.

[6] S. R. Qasim, J. Kieseler, Y. Iiyama, and M. Pierini, "Learning representations of irregular particle-detector geometry with distance-weighted graph networks," *Eur. Phys. J. C*, vol. 79, no. 7, p. 608, 2019.

[7] S. H. Rezatofighi, R. Kaskman, F. T. Motlagh, Q. Shi, D. Cremers, L. Leal-Taixé, and I. Reid, "Deep Perm-Set Net: Learn to predict sets with unknown permutation and cardinality using deep neural networks." arXiv:1805.00613, 2018.

[8] Y. Zhang, J. Hare, and A. Prügel-Bennett, "Deep Set Prediction Networks." arXiv:1906.06565, 2019.

[9] J. Duarte *et al.*, "Fast inference of deep neural networks in FPGAs for particle physics," *JINST*, vol. 13, no. 07, p. P07027, 2018.