Accelerating HEP research:
A Deep Learning framework

Francesco Vaselli

Pisa, 22/02/2022

Introduction

The need for fast and reliable computing methods in the physical sciences, notably in the high-energy field, has fueled much of the technological progress of modern times. Despite this, the average physicist has generally little time to spend on innovating and improving its computing toolkit, and has to content himself with boilerplate solutions. This is especially true in highly specialized fields, such as HEP, which purse a wide variety of research directions.

Specifically, in recent years, *machine learning* techniques have been massively adopted by scientific collaborations around the world. In particular, the paradigm known as *deep learning*, which leverages multiple layers of *artificial neurons* (theorized by [13]) trained through the use of a *loss function* and *backpropagation*, has achieved a wide range of successful usages. However, such tools remain geared towards the necessities of industry; much work remains to be done to enable the use of this technologies in hard sciences.

As a physicist with a keen interest in this type of applications, while still retaining useful domain knowledge, I am convinced that I could achieve significant results by pursuing the following research directions.

Research topics

What follows is a series of deep learning tools addressing the necessities of Run 3/ High Luminosity LHC: the HL-LHC will produce more than 250 inverse femtobarns of data per year and will be capable of collecting up to 4000 inverse femtobarns, making it possible to set new constrainst on a whole set of processes regarding the Higgs boson, such as its coupling to fermions of the second generation or the Higgs self-coupling. The vast amount of data will require efficient and fast *simulation techniques*, improvements in both *tracking* and *vertex reconstruction*, as well as reliable *trigger* and *real time* analysis frameworks.

Flash sim

This is the topic that I am currently pursuing with my Master thesis. The key idea is to directly generate a high level analysis format, such as CMS NANOAOD [12], training on fully simulated events. As a benchmark to evaluate the performance of such a simulation, the search data for the decay of Higgs to muon pairs in the VBF channel has been chosen. This kind of analysis requires only a limited number of muons and jets features to be simulated while still depending upon proper handling of correlations, so it is a good benchmark for a first prototype of this deep learning based approach. The goal of this kind of simulation, that we call *flash sim*, is to generate the full detector response (simulation and reconstruction) in a negligible time compared to a full simulation, hence enabling the generation of future large datasets at low computing cost.

Generative Adversarial Networks (GANs) [6] and *Variational Autoencorders* (VAEs) [9] are deep generative models which showed remarkable results in the field of computer vision, and have already been extensively investigated by the collaboration at CERN (see [2] and [10]); despite this, there is still a limited literature regarding behavior in low dimensionality as in our case, e.g. [8]. This models are also prone to *mode collapse* and failure to converge, and necessitate careful and domain-informed tuning to result in successful HEP applications.

A novel and promising approach is the one offered by Normalizing Flows [11], a family of methods for constructing flexible learnable probability distributions, often with neural networks, which allow us to surpass the limitations of simple parametric forms to represent complex high-dimensional distributions. In this case, a simple multivariate source of noise, for example a standard i.i.d. normal distribution, $X \sim \mathcal{N}(\mathbf{0}, I_{D\times D})$, is passed through a vector-valued invertible bijection, $g : \mathbb{R}^D \to \mathbb{R}^D$, to produce the more complex transformed variable $Y = g(X)$, and we can compose such bijective transformations to produce even more complex distributions, i.e. $Y = (g_{(0)} \circ g_{(1)} \circ \cdots \circ g_{(L-1)})(X)$.

The main challenge is in designing parametrizable multivariate bijections that have closed form expressions for both g and g^{-1}, a tractable Jacobian whose calculation scales with $O(D)$ rather than $O(D^3)$, and can express a flexible class of functions. Recent advancements have demonstrated the suitability of *spline transforms* (see [5]), and my Master thesis showed promising results in the generation of samples which provide convincing distributions as well as correct correlations and conditioning on the simulation ground truth.

Still, lots of potential application and improvements of this architecture for both generation and inference (see [7]) could be investigated.

Deep vertex and tracking

Possibly considered the holy grail of ML for HEP applications at the LHC, the problem of *track reconstruction* is mostly a pattern recognition task whose complexity grows much more than linearly with the increase of number of collisions, hence the number of tracks, and is thus expected to be one of the main problems for HL-LHC. When investigating such a large feature space, a common approach is to reduce the complexity of the network while still retaining useful information about local features through the use of *Convolutional Neural Networks* (CNNs), which nowadays are the standard approach for image datasets (as pioneered in [14]). Unfortunately, any representation of tracking detector as images would look very sparse and would not benefit of the locality idea of the CNN.

Another hard HEP problem is the reconstruction of secondary vertices used for b-tagging. In order to identify b-jets, a useful feature is the presence of the so called *secondary vertices* in a jet, i.e. points in space, displaced from the interaction point, where a group of tracks appears to originate from. A deep learning approach named *Deep Vertex* already exists, but it is implemented with CNN, thus limiting itself to a specific input format.

A promising way to overcome these challenges is the introduction of *Graph Networks* (GNs), which represent input and output data as graphs to exploit any invariance of the graph itself in order to perform the dimensionality reduction that is achieved in CNN. In the case of a GN, the locality is not based on an euclidean metric like as before, but rather on the number of connections needed to reach one node from another. In order to enforce this kind of locality, a *Message Passing* schema is often used for GN: the computation happens in several iterations where each iteration propagates information to/from neighbor nodes. Even a simple overview of the method goes beyond the scope of this section: we will now comment briefly on specific HEP advantages of this architecture, referring the reader to the seminal work of Battaglia *et al.* [1] for a comprehensive review.

The key idea behind GNs in HEP is to represent the data as a graph where the nodes are the hits (i.e. the individual measurements from tracking sensors) and nodes of subsequent layers are connected (with some pruning of nonphysical connections). The output is instead a graph made of several disconnected branches each representing an individual track (or track seed). We would like to emphasize how a *graph input topology* would possibly benefit many other models already in use at LHC, allowing a better application simply by performing a preprocessing step through the use of GN layers.

Due to the novelty of the approach, I could first deepen my knowledge of such methods this spring during my period as visiting student at CMS, acquiring the resources needed to start addressing real problems at the beginning of the PhD.

Trigger and real time analysis

Field-programmable gate arrays (FPGAs) have been used for decades to provide fast computing solutions, and are a cornerstone of modern trigger implementations. Their programming is traditionally done in Verilog/VHDL, i.e. low-level hardware languages: it si usally possible to translate C to Verilog/VHDL using High Level Synthesis (HLS) tools. Recent efforts have demonstrated the advantages of Deep Learning approaches to the problem of triggering, and produced a tool know as *hls4ml* [4] for the compression and implementation of Neural networks through the use of FPGAs.

non so bene che altro scrivere

Various researchers at the Pisa INFN section are already working extensively of FPGAs and trigger systems, providing the ideal ground for and early training and subsequent testing of ML methods for triggering during the rest of my PhD.

Quantum Machine Learning for HEP

Finally, a novel discipline born from *Quantum Computing* and ML, known as *Quantum Machine Learning* (QML) is already under serious investigation from the scientific community, due to the potential and significant *quantum* advantages. The limits of what machines can learn have always been defined by the computer hardware we run our algorithms on—for example, the success of modern-day deep learning with neural networks is enabled by parallel GPU clusters.

Quantum machine learning extends the pool of hardware for machine learning by an entirely new type of computing device—the *quantum computer*. Some research focuses on ideal, universal quantum computers ("fault-tolerant QPUs") which are still years away. But there is rapidly-growing interest in quantum machine learning on near-term quantum devices (*Noisy Intermediate-Scale Quantum* or NISQ).

We can understand these devices as special-purpose hardware like Application-Specific Integrated Circuits (ASICs) and Field-Programmable Gate Arrays (FPGAs), which are more limited in their functionality but nonetheless well suited to specific applications. In the modern viewpoint, quantum computers can be used and trained like neural networks. We can systematically adapt the physical control parameters, such as an electromagnetic field strength or a laser pulse frequency, to solve a problem. Additionally, quantum circuits are differentiable, and a quantum computer itself can compute the change in control parameters needed to become better at a given task. Trainable quantum circuits can be leveraged in other fields like quantum chemistry or quantum optimization. It can help in a variety of applications such as the design of quantum algorithms, the discovery of quantum error correction schemes, and the understanding of physical systems.

At the moment the effort of the HEP community is focused on quantum generative models (see [3]), where the noisy behavior is mitigated or even beneficial. The CERN has already established a pertnership with IBM, a leading competitor for quantum technologies, and founded the CERN Quantum Technology Initiative (CERN QTI), a comprehensive R&D, academic and knowledge-sharing initiative to exploit quantum advantage for high-energy physics and beyond.

As a novel and exciting field, I would have lots of possible research direction, starting from possible extension of my Master thesis work and possibly developing entirely novel approaches. Both the SNS, with its dedicated courses in Quantum Information, and the INFN, a member of various LHC collaborations, would serve as the perfect starting point for this direction of research.

References

[1] Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Caglar Gulcehre, Francis Song, Andrew Ballard, Justin Gilmer, George Dahl, Ashish Vaswani, Kelsey Allen, Charles Nash, Victoria Langston,

Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matt Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational inductive biases, deep learning, and graph networks, 2018.

[2] Anja Butter, Tilman Plehn, and Ramon Winterhalder. How to gan lhc events. *SciPost Physics*, 7(6), Dec 2019.

[3] Su Yeon Chang, Sofia Vallecorsa, Elías F. Combarro, and Federico Carminati. Quantum generative adversarial networks in a continuous-variable architecture to simulate high energy physics detectors, 2021.

[4] J. Duarte, S. Han, P. Harris, S. Jindariani, E. Kreinar, B. Kreis, J. Ngadiuba, M. Pierini, R. Rivera, N. Tran, and Z. Wu. Fast inference of deep neural networks in fpgas for particle physics. *Journal of Instrumentation*, 13(07):P07027–P07027, Jul 2018.

[5] Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural spline flows, 2019.

[6] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[7] Stephen R. Green and Jonathan Gair. Complete parameter inference for gw150914 using deep learning, 2020.

[8] Felix Jimenez, Amanda Koepke, Mary Gregg, and Michael Frey. Generative adversarial network performance in low-dimensional settings, 2021-04-20 04:04:00 2021.

[9] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2014.

[10] Sydney Otten, Sascha Caron, Wieske de Swart, Melissa van Beekveld, Luc Hendriks, Caspar van Leeuwen, Damian Podareanu, Roberto Ruiz de Austri, and Rob Verheyen. Event generation and statistical sampling for physics with deep generative models and a density information buffer, 2021.

[11] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows, 2016.

[12] Andrea Rizzi, Giovanni Petrucciani, and Marco Peruzzi. A further reduction in CMS event data for analysis: the NANOAOD format. In *European Physical Journal Web of Conferences*, volume 214 of *European Physical Journal Web of Conferences*, page 06021, July 2019.

[13] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65 6:386–408, 1958.

[14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.