IL NUOVO CIMENTO **46 C** (2023) 97
DOI 10.1393/ncc/i2023-23097-6

COMMUNICATIONS: SIF Congress 2022

Innovative deep neural networks resizing for FPGA implementation in future collider experiments

D. MASCIONE(1)(2)(3), M. CRISTOFORETTI(2)(3), A. DI LUCA(2)(3), F. M. FOLLEGA(1)(3) and R. IUPPA(1)(3)

(1) *Dipartimento di Fisica, Università di Trento - Via Sommarive 14, 38123 Povo (TN), Italy*
(2) *Fondazione Bruno Kessler - Via Sommarive 18, 38123 Povo (TN), Italy*
(3) *TIFPA-INFN - Via Sommarive 14, 38123 Povo (TN), Italy*

received 31 January 2023

Summary. — Future collider experiments are expected to produce thousands of ExaBytes of data per year, and it will become impractical to maintain old trigger schemes to preserve data for offline analysis. Deep Neural Networks offer a great opportunity to take fast and accurate decisions, but it is essential to employ Deep Learning algorithms online at the early stage of event selection, taking advantage of FPGAs. In order to optimize Deep Neural Networks under size limits to accommodate resources of FPGAs, we developed a novel pruning technique. The proposed method can reduce the overall sizes of neural networks by pruning unnecessary nodes, with the network final dimensions determined by the user.

1. – Introduction

The generation of large amounts of data, which must be properly managed, represents one of the main issues in collider experiments. In one year of operation, approximately 40k ExaBytes of raw data are produced at the Large Hadron Collider (LHC) at CERN [1]. By 2026, it is believed that the amount of data produced will equal 800 PB per year, thanks to the High Luminosity LHC (HL-LHC) upgrade. The raw data are currently filtered by ATLAS and CMS (the two main general-purpose particle detectors at the LHC) using a two-step selection process [2]. The selection process of relevant events, known as triggering, allows for storing salient data for later analysis. In the first stage of selection, the Level-1 trigger, the event rate is decreased making use of algorithms implemented on Field Programmable Gate Arrays (FPGAs). Potentially interesting data are then further processed using selection algorithms on easily accessible CPUs and GPUs in the so-called High-Level Trigger, the second stage of the triggering chain.

A significant problem is making sure that important information is not discarded, and the ever increasing amount of data produced makes it hard to maintain existing trigger schemes. Help in this sense can come from Deep Learning algorithms, which have proven to be extremely useful and effective when dealing with large amounts of data [3]. The High Energy Physics community has made several efforts to investigate the potential to apply Deep Learning in the selection stages, particularly in the Level-1 trigger, before any selection bias is introduced. Thanks to recent developments, Deep Neural Networks

(DNNs) can be implemented on the FPGAs installed on L1T boards in a relatively simple way [4], but must be modified to meet the available resources.

One way of modifying DNNs in this respect is represented by network pruning, which consists in removing uninteresting, redundant, or unneeded components. There exist several different approaches to pruning DNNs, but they often present some limitations and it is difficult to find an ideal criterion that ensures good performance while respecting the size constraints posed by FPGAs. In this context, we developed a novel pruning technique for reducing the size of DNNs by removing unnecessary nodes under size constraints. Our method is versatile and simple to implement into already developed Deep Learning models and allows selecting the best-performing network architecture suitable to the FPGA resources available, by automatically identifying the most useful nodes of each layer while strictly adhering to the final size constraint.

2. – Pruning Deep Neural Networks

DNNs are computational models composed of multiple layers of interconnected basic computational units, called artificial neurons or nodes [5]. Their role is to convert weighted inputs into outputs in a process called training. During the training process, the model learns to recognize a pattern in the input data by adjusting the weights of the connections between neurons in the different layers. Successful DNNs typically require vast amounts of computation and memory to complete their learning task, and the final trained model is not always compatible with the programmable resources of FPGAs (such as the number of logic units and memory slots). Before being implemented on FPGAs, DNNs must therefore be properly optimized. Pruning, which consists in removing unimportant components from an existing network with a minimal impact on performance, is a widely used strategy to decrease the amount of resources required by DNNs [6].

There are numerous approaches to pruning a DNN [7]. Nearly all neural network pruning algorithms use a similar procedure according to which the network is first trained to convergence. After that, a score is assigned to each parameter or structural element in the network, and the network is pruned based on these scores. Pruning diminishes the network's accuracy, therefore it must be retrained (a practice known as fine-tuning) to recover. The pruning and fine-tuning process is frequently repeated numerous times, progressively shrinking the network's size.

Pruning can be applied to different DNN components, including individual connections, nodes, and even entire layers. Currently used pruning schemes mainly focus on removing single connections. This represents a quick and easy way for reducing the size of a model. However, such a pruning strategy result in a sparse network structure, which could be hard to train [8]. On the other hand, acting on neurons or layers does not harm the network structure while producing a smaller, faster, and more resource-efficient DNN.

Pruning single connections after training followed by one or more fine-tuning campaigns is a popular pruning approach. However, this strategy can be time-consuming and may not always result in the most convenient choice. Based on this and the aforementioned considerations, we looked into a different method of reducing the size of DNNs. Our strategy is focused on removing unnecessary nodes, rather than individual connections. Pruning occurs in one shot during the training stage and there is no need for a separate fine-tuning campaign after pruning. Moreover, the final dimension of the pruned network can be determined by the user. This method works by layering a shadow network on top of the DNN that has to be optimized. The neurons of this shadow network have just one connection to each single node of the original network and act as a filter:

the calculations made by the shadow nodes during training are such that their output will be zeroed when they are connected to "unwanted" nodes. As a result, a portion of the neurons will only be used for learning, and certain nodes will be "turned off". The training is optimized for learning with the precise number of nodes required by the user, while which neurons will be selected for learning is determined automatically by the shadow network. This method enables pruning the whole DNN as well as just a portion of it, depending on where the shadow network is set.

A high-quality pruning that actually adheres to the planned global compression rate can be accomplished with this innovative pruning approach. Our strategy has also been shown to be helpful in selecting the best model architecture from a range of options that are appropriate for the available device resources, and can be adopted for future FPGA implementation of DNNs.

3. – Test results

A DNN designed to recognize jets that contain b-quarks originating from boosted Higgs boson decay in proton-proton collision events has been resized using the aforementioned pruning method. Since the $H \to b\bar{b}$ channel is responsible for 58% of all Higgs boson decays [9], it is very important for the study of the Higgs boson properties. However, in a proton-proton collision experiment, it can be challenging to correctly identify these events due to the enormous, irreducible background caused by QCD multi-jet generation. The DNN utilized in the testing consisted of 4 hidden layers of 64 nodes each for 40 input variables, and was created to separate this background from the Higgs boson decay without taking pile-up effects into account.

The same original DNN has been pruned in independent trainings changing the amount of desired nodes to be used for learning. The results are shown in fig. 1, where the background rejection rate is represented as a function of the Higgs tagging efficiency. Better DNN performance is indicated by a higher rate of background rejection for each tagging efficiency value. As shown in the plot on the left, requiring a higher number of active nodes leads to better performance as expected; this suggests that only the given percentage of nodes are being used for learning, while the remaining nodes have been excluded from training.

Another interesting result is shown in the plot on the right of fig. 1. Once pruning is finished, the shadow network can be removed together with the unnecessary nodes of the original network. This leads to a reduced DNN with a well-defined architecture, that can be trained as a new independent model. Tests have shown that the obtained DNN performance is consistent with the one obtained by pruning the original network during the training stage. This confirms that the nodes actually used for the learning task are the ones selected by our method, and supports the idea that combining training and pruning into a single action can be an interesting alternative to the traditional training-pruning-retraining pipeline. In fact, since our strategy does not call for an initial training phase prior to pruning or a final fine-tuning of the model, it actually offers a quick method for resizing DNNs.

4. – Conclusion

Future collider experiments will generate enormous amounts of data, making it crucial to exploit the possibility of implementing DNNs on FPGAs for selecting interesting events in real-time. Deep Neural Networks represent a fantastic tool for such a challenging task,

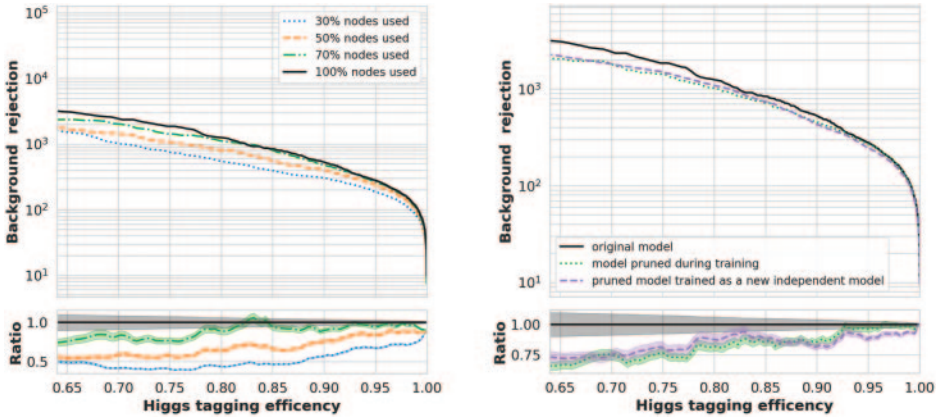

Fig. 1. – Background rejection versus Higgs tagging efficiency for models pruned during training with different compression rates required (on the left). Background rejection versus Higgs tagging efficiency for the original model, the model pruned during training and the pruned model trained as a new independent DNN (on the right) obtained with the 70% of nodes of the original model.

but must be suitably optimized for FPGA implementation. We created a novel pruning technique to resize DNN within the constraints imposed by the limited resources of FPGAs. With our method, the user can choose the final network dimensions, while the overall size of the neural network is shrunk by removing superfluous nodes. Pruning is performed in a fast way together with the training itself. This procedure does not require any fine-tuning and has shown interesting results, which could lead to future use in high-energy physics experiments.

* * *

The work described in this paper has been carried out in a joint effort by the members of the *deepPP* initiative of the University of Trento and Fondazione Bruno Kessler. For contacts and information about the group's activities, please visit `https://www.deeppp.eu/`.

REFERENCES

[1] Clissa L., arXiv:2202.07659 (2022).
[2] Smith W. H., *Ann. Rev. Nucl. Part. Sci.*, **66** (2016) 123.
[3] Najafabadi M. M. *et al.*, *J. Big Data*, **2** (2015) 1.
[4] Duarte J. *et al.*, *JINST*, **13** (2018) P07027.
[5] LeCun Y *et al.*, *Nature*, **521** (2015) 436.
[6] Cheng Y. *et al.*, *IEEE Signal Process. Mag.*, **35** (2018) 126.
[7] Blalock D. *et al.*, *Proc. Mach. Learn. Syst.*, **2** (2020) 129.
[8] Evci U. *et al.*, arXiv:1906.10732 (2019).
[9] LHC Higgs Cross Section Working Group collaboration, *Handbook of LHC Higgs Cross Sections: 4. Deciphering the Nature of the Higgs Sector*, in *CERN Yellow Reports: Monographs* 2/2017 (2017).