Frascati Physics Series Vol. 72 (2022)
7TH YOUNG RESEARCHERS' WORKSHOP 'PHYSICS CHALLENGES IN THE LHC ERA'
July 11 and 14, 2022

PRUNING DEEP NEURAL NETWORKS FOR LHC CHALLENGES

Daniela Mascione
Università degli Studi di Trento and TIFPA, Via Sommarive 14, 38123 Trento, Italy
Fondazione Bruno Kessler, Via Sommarive 18, 38123 Trento, Italy

Abstract

The enormous amount of data generated at the Large Hadron Collider makes it challenging to maintain current event selection mechanisms. There is growing interest in attempting to make use of Deep Neural Networks for event selection with the aid of FPGAs, already employed at the selection early stages. However, because of the constraints imposed by systems based on FPGAs, Deep Learning algorithm design is made more difficult. We therefore investigated a pruning strategy for quickly optimizing Deep Neural Networks under size constraints to fit the resources of FPGAs.

1 Introduction

The Large Hadron Collider (LHC) at CERN produces on average 40 million proton-proton collision events each second. As a result of the detection of the particles produced in these events in the sensors of detectors positioned all around the LHC ring, roughly 40k ExaBytes of raw data are generated in one year of operation [1]. Due to bandwidth restrictions, the main general-purpose particle detectors at the LHC, ATLAS and CMS, discard the majority of collision events through a two-steps selection mechanisms [2]. The initial selection stage, known as the level-1 trigger (L1T), is where the majority of events are discarded. Its job is to reduce the event rate by 2 orders of magnitude in a few microseconds ($\mathcal{O}(1)\mu$s). In the L1T algorithms are implemented as programmable logic on special electronic boards with field-programmable gate arrays (FPGAs). The events acknowledged by the L1T are then further processed in the so-called High Level Trigger (HLT) with selection algorithms on readily available CPUs and GPUs.

Making sure not to discard interesting events is a big challenge, and Deep Learning algorithms might be useful in this regard. There are several community efforts to explore the possibility to apply Deep Learning in the selection stages, especially in the L1T, prior to the introduction of any selection

bias. Recent developments in this field make it possible to deploy Deep Neural Networks (DNNs) on the FPGAs mounted on the L1T boards [3]. However, DNNs have to be adapted to fit the L1T infrastructure. In this context, we investigated an effective method to resize DNNs by pruning superfluous nodes.

2 Deep Neural Networks

DNNs are computing systems designed for non-linear learning problems [4]. They are based on a collection of connected basic units or nodes called artificial neurons that are aggregated into layers. Simple neural network architectures are made of three kinds of layers: the input layer, the output layer, and the hidden layer. Networks that have more than one hidden layer are called Deep Neural Networks.

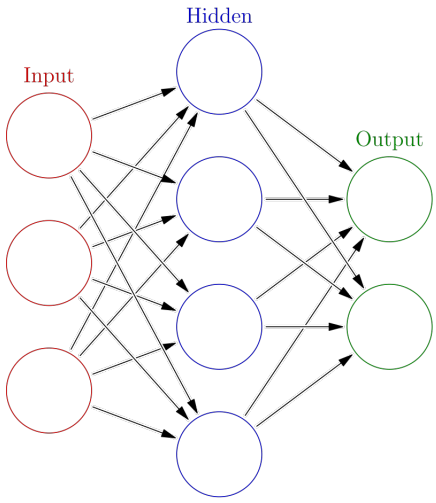

Figure 1: *Schematic representation of an artificial neural network with the input layer, the output layer, and one hidden layer. Source: https://commons.wikimedia.org/wiki/File:Colored_neural_network.svg*

Through a process called training, neural networks learn to recognize a pattern in the input data. During training, nodes convert weighted inputs into outputs: each neuron performs calculations that comprise a linear combination of the input data, which is then passed through a non-linear function called activation function. The activation function's primary goal is to disrupt the model's linearity. All these computations are carried out over the entire network: neurons receive inputs and produce a single output that is sent to multiple other neurons of subsequent layers. The inputs to a neuron may be the outputs of other neurons or they may be external data (such as images). The goal of recognizing a pattern in the input data (such as identifying an object in an image, for example) is accomplished by the outputs of the neural network's final output neurons (that may, for instance, be the likelihood of an object appearing in an image). The network's performance is then evaluated in relation to the expected output and the network's various parameters are modified to restart the calculation process in order to achieve an accurate result.

3 Pruning

Typically DNNs, especially the more effective ones, call for colossal amounts of computation and memory. These demands don't always correspond to the FPGAs' programmable resources (like the number of logic

units and memory slots), hence DNNs must be optimized before being implemented on FPGAs. Neural network pruning, which consists in eliminating superfluous structures from an existing network, is a popular strategy for lowering DNNs resource requirements [5]. The goal is to downsize a large, accurate starting network without suffering too much performance loss.

There are many different strategies to prune a DNN. The most popular pruning techniques are based on removing single parameters in accordance with a particular ranking determined after the starting network has been trained to convergence [6]. The pruned model is then retrained to recover from performance loss. Typically, pruning and retraining are conducted repeatedly, gradually shrinking the network. These techniques can be time-consuming as a result. For this reason, we investigated an alternative strategy for shrinking DNNs by removing during training the number of nodes determined by the user. This strategy works by adding a shadow network - whose neurons have just one connection to each of the single nodes of the original network - on top of the DNN that needs to be optimized. The layers of the shadow network contribute to training, as training is optimized for learning with precisely the required number of nodes: the calculations performed by the shadow nodes during training are such that their output will be zeroed when they are connected to "undesired" nodes. As a result, some nodes will be "switched off" and only a fraction of neurons will be used for learning.

4 Tests and results

The aforementioned pruning strategy has been tested to resize a DNN used to identify jets that contain b-quarks originating from boosted Higgs bosons decay in proton-proton collision events. The $H \to b\bar{b}$ channel accounts for 58% of all Higgs boson decays [7], and it is therefore important for the investigation of Higgs boson properties. However, it can be difficult to identify these events in a proton-proton collision experiment because of the massive, irreducible background coming from QCD multi-jet production. The DNN used for tests was developed to distinguish between this background and the Higgs boson decay, without including pile-up effects.

Different networks were pruned during training by varying the number of desired nodes to be used for learning. Figure 2 shows the background rejection rate as a function of the Higgs tagging efficiency. A higher rate of background rejection for each tagging efficiency value denotes better DNN performance. Better performance is achieved with higher number of nodes required, as expected: this suggests that only the indicated percentage of nodes is actually used for learning, while the remaining nodes have been "turned off".

In theory, the pruned DNNs might be retrained as independent models. The results are consistent with those that can be achieved by pruning during training, as shown in Figure 3, and there is therefore no need for fine-tuning following pruning.

5 Conclusion

A pruning strategy for reducing the number of DNNs nodes used for learning has been investigated. As a result, the overall size of the neural network is reduced, with the user ultimately determining its final dimensions. This makes it possible to adapt Deep Neural Networks to fit the resources of FPGAs used at the early stages of event selection at the LHC, in the challenging task of making sure not to discard interesting events.

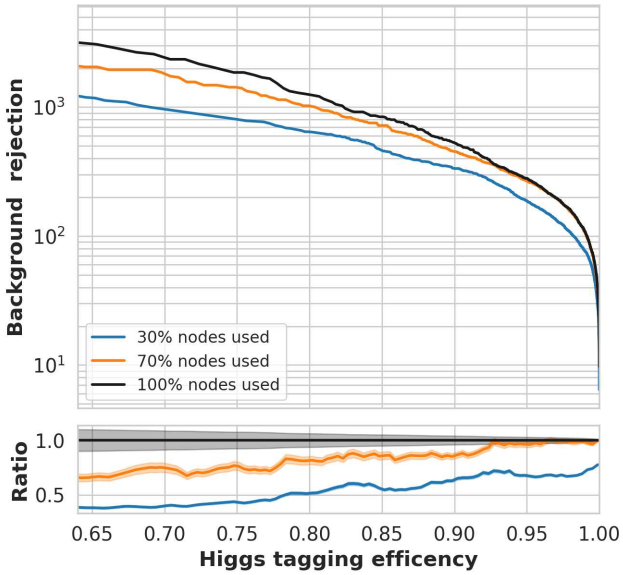

Figure 2: *Background rejection rate versus Higgs tagging efficiency for different models pruned during training by varying the number of desired nodes.*

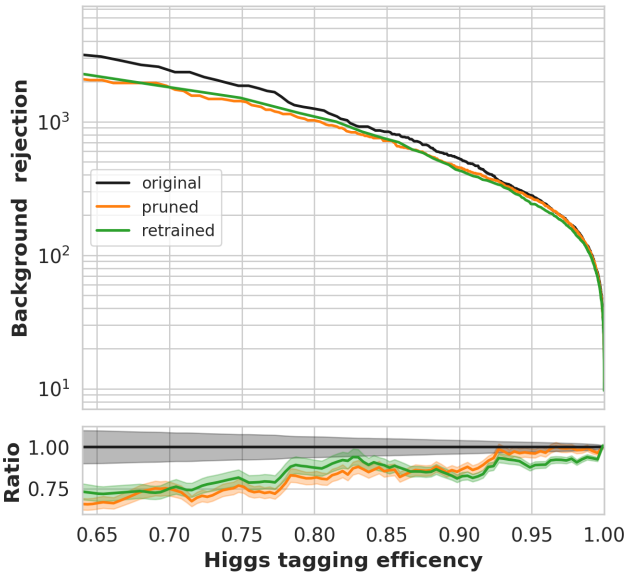

Figure 3: *Background rejection rate versus Higgs tagging efficiency for the original model (black), the model pruned during training (orange) and the retrained pruned model (green).*

6 Acknowledgements

The work described in this paper has been carried out in a joint effort with Andrea Di Luca, Francesco Maria Follega, Marco Cristoforetti and Roberto Iuppa, members of the *deepPP* group 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. L. Clissa, arXiv:2202.07659 (2022).
2. W.H. Smith, Annu. Rev. Nucl. Part. Sci., **66**, 123 (2016).
3. J. Duarte *et al.*, JINST **13**, P07027 (2018).
4. Y. LeCun *et al.*, Nature **521**, 436 (2015).
5. Y. Cheng, *et al.*, IEEE Signal Processing Magazine, **35 (1)**, 126 (2018).
6. D. Blalock *et al.*, What is the state of neural network pruning?, in: Proceedings of machine learning and systems 2, 129 (2020).
7. 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).