Fast inference using FPGAs for DUNE data reconstruction

Manuel J. Rodriguez[1,*] for the DUNE Collaboration

[1]CERN, Genève, Switzerland

Abstract. The Deep Underground Neutrino Experiment (DUNE) will be a world-class neutrino observatory and nucleon decay detector aiming to address some of the most fundamental questions in particle physics. With a modular liquid argon time-projection chamber (LArTPC) of 40 kt fiducial mass, the DUNE far detector will be able to reconstruct neutrino interactions with an unprecedented resolution. With no triggering and no zero suppression or compression, the total raw data volume would be of order 145 EB/year. Consequently, fast and affordable reconstruction methods are needed. Several state-of-the-art methods are focused on machine learning (ML) approaches to identify the signal within the raw data or to classify the neutrino interaction during the reconstruction. One of the main advantages of using those techniques is that they will reduce the computational cost and time compared to classical strategies. Our plan aims to go a bit further and test the implementation of those techniques on an accelerator board. In this work, we present the accelerator board used, a commercial off-the-shelf (COTS) hardware for fast deep learning (DL) inference based on an FPGA, and the experimental results obtained outperforming more traditional processing units. The FPGA-based approach is planned to be eventually used for online reconstruction.

1 Introduction

The Deep Underground Neutrino Experiment (DUNE) will be an international neutrino observatory designed to answer fundamental questions about the nature of elementary particles and their role in the universe [1]. The DUNE far detector (FD) will be located about 1.5 km underground at the Sanford Underground Research Facility (SURF) in South Dakota, US, at a distance of 1300 km from Fermilab where the world's most intense neutrino beam will target the FD. The FD will be composed of four liquid argon time-projection chambers (LArTPC) each of them with a total fiducial mass of 10 kt. The liquid-argon technology allows us to reconstruct neutrino interactions with image-like precision and unprecedented resolution.

2 The DUNE data challenge

The data acquisition (DAQ) system for the DUNE FD gathers beam-related interactions, as well as cosmic-ray muons and atmospheric neutrino interactions; added together, recording their activity will dominate the data rate. Before triggering, the data rate for each 10-kt module is expected to be as much as 1.5 TB/s. The ultimate limit on the output data rate of

[*] e-mail: mjrodrig@cern.ch

the DAQ is set by the available permanent storage capacity; this limit is estimated to be about 30 PB/year. Extrapolating to four detector modules, this requires a DAQ data reduction factor of almost four orders of magnitude. In order to meet these demands, new technologies will need to be developed, including high throughput front-end electronics as well as additional FPGA and CPU resources.

Deep learning (DL) techniques, such as deep neural networks (DNN) or convolutional neural networks (CNN), have demonstrated to be extremely useful in particle physics experiments [2–4], also in neutrino experiments [5, 6]. However, standard computing infrastructure, i.e., CPUs, is usually not suitable for this ever-increasing technology, so other concrete solutions are needed.

2.1 Machine learning on hardware accelerators

There is a growing demand for computing resources needed by modern machine learning (ML) methods; consequently, hardware accelerators have entered in place. Nowadays, we can find all kinds of accelerators, from general-purpose computing units, such as standard GPUs [7], to specialized devices designed to speed up ML workloads [8].

The use of field-programmable gate arrays (FPGA) plays a crucial role in hardware accelerators. Programming custom logics directly on the chip allows us to obtain maximum performance from the hardware without needing to manufacturing an application-specific integrated circuit (ASIC). As a disadvantage, FPGAs are generally challenging to program, and their capacity remains very limited, but this is changing in the last years [9].

The high-level synthesis (HLS) language introduces a more intuitive way for even non-experts to program FPGAs in a C/C++ like code [10]. Some techniques allow to convert neural networks to HLS in a quasi automated way [11]. The work that we present is an efficient way to implement DNN, especially CNN, into FPGAs, avoiding the complex part of hardware programming.

3 The Micron Deep Learning Accelerator technology

The Micron Deep Learning Accelerator (DLA) is a FPGA-based unit from Micron (SB852) that has been designed for running neural networks with high efficiency, high speed, low power consumption and low latency even with small batches. It has a Xilinx Virtex Ultrascale+ UV9P FPGA, 64 GB of DDR4, 2 GB of HMC memory, 2 QSFP transceiver connectors and a PCIe x16 Gen3 interface. The FPGA contains a custom firmware that turns the FPGA into a dedicated processor, with 2 clusters (cores) containing 1024 MAC units each. The MACs are divided among various sub-units (matrix-matrix, matrix-vector and vector-vector) with several parallel connections to internal maps (2MB/cluster) and kernel (512KB/cluster) buffers and the memory interface for optimal access to memory. All operations are performed on 16-bit fixed points values with intermediate results kept in a 32-bit accumulator. This implies a reduction in precision compared to floating-point that has to be considered when designing and deploying neural networks.

The DLA comes with a complete framework that allows quick deployment of existing neural networks designed with common deep learning frameworks like Pytorch, TensorFlow, Keras and others. The Micron SDK has a compiler that will compile networks exported to ONNX (a common neural networks interchange format) into a binary code that the accelerator can run. The compiled code will stay in the accelerator DDR4 memory, which is shared between the FPGA and the host, so different networks can be quickly switched on the accelerator, without programming a newer firmware onto the FPGA. Examples are provided with both C and Python code and turning a CPU or GPU based code into a Micron accelerator code takes just a few lines of code of modification.

4 The DUNE Convolutional Visual Network

The DUNE Convolutional Visual Network (CVN) [12, 13] is an algorithm for identifying neutrino interactions based on their topology and without the need for detailed reconstruction algorithms. In general terms, it is a CNN, inspired by the ResNet-18 architecture [14]. This paper aims to demonstrate that we can implement the CVN on the Micron DLA. Similar techniques have been demonstrated to outperform traditional reconstruction methods in high energy physics [15].

The DUNE CVN takes 500x500x3 pixel images of the neutrino interactions as input. These images are produced by concatenating three 500x500x1 pixel images - one from each readout view of the DUNE LArTPCs (Figure 1) - along the third dimension (RGB channels). The images contain the charge and the peak time of the reconstructed hits and do not use any information beyond the hit reconstruction.

The primary goal of the DUNE CVN is to efficiently and accurately produce event selections of the neutrino interactions. We consider thirteen categories:

- For charged-current (CC) interactions, and for each of the neutrino flavors, CC ν_μ, CC ν_e and CC ν_τ: CC quasi-elastic (CC QE), CC resonant (CC Res), CC deep inelastic (CC DIS) and CC other.
- Neutral current (NC).

Once the DUNE CVN is trained, it returns scores for each event to be in the above thirteen categories; the thirteen scores sum to 1, meaning that each value gives a fractional score that can be used to classify images. However, during the analysis, we sum together the scores of the four sub-categories for each neutrino flavor. This is done because the DUNE analysis is focused on the CC ν_μ and CC ν_e selections.

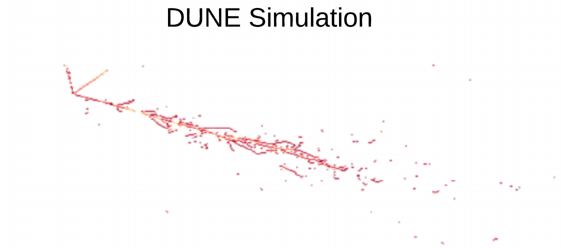

Figure 1: A simulated 2.2 GeV ν_e CC interaction shown in the collection view of the DUNE LArTPCs. The horizontal axis shows the wire number of the readout plane and the vertical axis shows time. The greyscale shows the charge of the energy deposits on the wires. The interaction looks similar in the other two views. [13]

The DUNE CVN was trained using approximately 3 million neutrino interactions from a Monte Carlo simulation that are independent of the sample that is used to generate the physics measurement sensitivities. Since the DUNE analysis is focused on CC ν_μ and CC ν_e, the sample was chosen to ensure similar numbers of training samples from the two aforementioned flavors.

5 Benchmark

In this section, we will describe the benchmark ran consisting of three independent tests to characterize the Micron DLA. Since the performance of the DUNE CVN is already known and described in [12], we aimed to check whether we could have the same results using the Micron DLA and obtain an increase of performance. For this purpose, we tested the DLA on three different scenarios, using the DUNE CVN for all of them.

For the first test, we ran inference continuously over ~2 million images, using the SB852. Then we compared the results with the ground truth. Table 1 shows the classification report. To fully understand the table, some metrics have to be defined. We define $C_{i,j}$ as the number of elements predicted as category i actually belonging to the category j with $i, j = 1, 2, ..., n$, where n is the number of categories:

Precision: it measures the number of correctly classified items in a category over all items predicted as this category.

$$Precision(i) = \frac{C_{i,i}}{\sum_{j=1}^{n} C_{i,j}} \tag{1}$$

Recall: is the number of correctly predicted elements in a category over the number of actual elements in the category.

$$Recall(i) = \frac{C_{i,i}}{\sum_{j=1}^{n} C_{j,i}} \tag{2}$$

F1-Score: it acts as a weighted average of precision and recall. The F1 score is limited between 0 and 1, where 0 is the worst value, and 1 is the best.

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{3}$$

Support: is the total number of elements, $\sum C_{i,j}\ \forall i$, for each category, j.

The results presented in Table 1 are the expected results for the DUNE CVN [12], proving that the NN performs correctly in the inference engine.

Table 1: Classification report for each of the 13 different categories.

Interaction	Precision	Recall	F1-score	Support
ν_μ CC QE	0.79	0.80	0.80	113213
ν_μ CC Res	0.59	0.67	0.62	157227
ν_μ CC DIS	0.70	0.77	0.73	203583
ν_μ CC other	0.71	0.24	0.36	54752
ν_e CC QE	0.78	0.79	0.79	110484
ν_e CC Res	0.61	0.70	0.65	154098
ν_e CC DIS	0.68	0.75	0.72	197268
ν_e CC other	0.59	0.43	0.50	54252
ν_τ CC QE	0.56	0.17	0.26	21447
ν_τ CC Res	0.42	0.06	0.10	23373
ν_τ CC DIS	0.50	0.29	0.37	46824
ν_τ CC other	0.49	0.05	0.09	10262
NC	0.91	0.94	0.92	773217

The set of $C_{i,j}$ values can be illustrated as a matrix, where the predicted categories, i, correspond to the rows and the actual labels, j, to the columns. This matrix is called “confusion matrix” and helps to interpret the reported results. Figure 2 shows the confusion matrix for

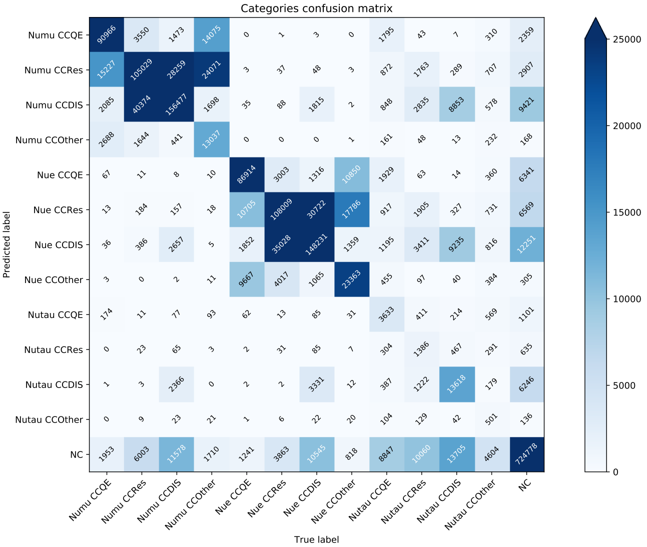

Figure 2: Confusion matrices for the classification task of the different neutrino interaction. A confusion matrix shows the number of elements, $C_{i,j}$, predicted as category i, in rows, belonging to the category j, in columns.

the classification report. The color scale of the matrix works the following way: the lightest color represents cells with no classified events, while the darkest color represents cells with more than 25k classified events. The elements in the main diagonal show the number of correctly predicted samples.

The highest values tend to cluster around the same neutrino flavor, and that is intrinsic to the neutrino interactions topology. It is easier to distinguish between neutrino flavors than interactions; therefore, sometimes the network mixes the different interactions within the same flavor. As mentioned in Section 4, the DUNE analysis is focused on the CC ν_μ and CC ν_e selections. The Table 2 shows the classification report after summing together the scores of the four sub-categories for each neutrino flavor. With an F1-score of 0.94 and 0.93 for CC ν_μ and CC ν_e, respectively, this network maximizes the sensitivity of the experiment for the neutrino classification analysis.

Table 2: Classification report for each neutrino flavor.

Interaction	Precision	Recall	F1-score	Support
CC ν_μ	0.93	0.95	0.94	528775
CC ν_e	0.89	0.96	0.93	516102
CC ν_τ	0.58	0.31	0.40	101906
NC	0.92	0.92	0.92	773217

For the second test, we reran the same network on a smaller dataset using 1,500 of randomly chosen images. This time, we deployed it on a NVIDIA Tesla V100 GPU and on the SB852 to compare their outputs. The goal is to check if there is any discrepancy due to the loss of precision due to the lack of floating-point arithmetic as mentioned in Section 3. Figure 3 shows the histogram of the absolute error for each of the outputs for all samples. With a standard deviation of 0.0416 and a mean in the order of magnitude of 10^{-10}, we can conclude that the loss of precision is negligible on this test.

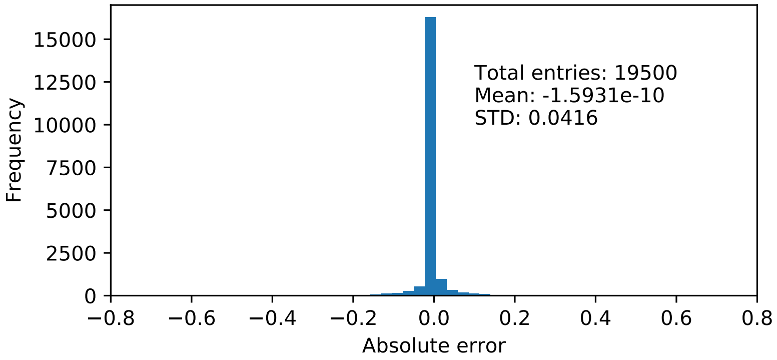

Figure 3: Histogram for the absolute error between the GPU and the SB852 board. The error is calculated as the difference in GPU and SB852 output for the 13 categories over a total of 1500 samples.

The aim of the third test carried out is to measure the performance of the SB852 compared to a traditional processor unit. For this test, we used an Intel Core i7-8750H 8th Gen CPU using the Keras framework with TensorFlow as backend. We enabled multithreading pools in TensorFlow to get the maximum performance of the CPU. On the SB852 side, we used the 4th Gen DLA firmware with 512 MACs running at 250MHz. We ran the inference on a loop of 145 samples and eliminated the first 20 iterations until we reached a steady state. Table 3 depicts the results. The average inference time in CPU is 264.85 ms. The SB852 is almost 2.6 times faster, with an average inference time of 103.61 ms.

Table 3: Inference time comparison between the Micron SB852 and in Intel CPU processor.

Processor	Average time (ms)	STD	Min	Max
SB852	103.6074	0.5505	102.4658	105.0381
CPU (i7-8750H)	264.8545	0.8653	262.1692	267.2548

6 Conclusion

In this work, we presented an efficient way to run a NN on FPGAs using the Micron DLA. Due to the amount of data that DUNE will produce per year, approaches that allow decreasing its volume are crucial for its smooth operation. We successfully implemented a NN conceived to classify neutrino interactions into the Micron DLA SB852. We tested its behavior over ~2 million images with a negligible error compared to its original implementation. Once we characterized the DLA for neutrino physics applications, we plan to move to a more detector-specific scenario, with extremely tight constraints where efficiency in data management and operation is critical. Machine learning techniques, such as DNN or CNN, can do the work, but only if they can be deployed efficiently on hardware accelerators that can meet these constraints.

Acknowledgements

This work was supported by Micron and CERN openlab. We would especially like to acknowledge Saúl Alonso-Monsalve and Leigh H. Whitehead for his work to the DUNE CVN together with his team. And to our CERN colleagues Paola Sala and Maurizio Pierini, as well as to all our colleagues at Micron: Jason Adlard, Mark Hur, Eugenio Culurciello, Marko Vitez, Andre Chan, Stuart Grime, Dustin Werran, Aliasger Zaidy, Abhishek Chaurasia.

References

[1] B. Abi, R. Acciarri, M.A. Acero, G. Adamov, D. Adams, M. Adinolfi, Z. Ahmad, J. Ahmed, J. Ahmed, T. Alion et al. (DUNE), *Deep Underground Neutrino Experiment (DUNE), Far Detector Technical Design Report, Volume 1 Introduction to DUNE*, in *Far Detector Technical Design Report, Volume 1 Introduction to DUNE* (2020), http://arxiv.org/abs/2002.02967

[2] C. Adam-Bourdarios, G. Cowan, C. Germain, I. Guyon, B. Kégl, D. Rousseau, *The Higgs boson machine learning challenge*, in *Proceedings of the NIPS 2014 Workshop on High-energy Physics and Machine Learning*, edited by G. Cowan, C. Germain, I. Guyon, B. Kégl, D. Rousseau (PMLR, Montreal, Canada, 2015), Vol. 42 of *Proceedings of Machine Learning Research*, pp. 19–55, http://proceedings.mlr.press/v42/cowa14.html

[3] F. Carminati, G. Khattak, M. Pierini, A. Farbin, B. Hooberman, W. Wei, M. Zhang, V.B. Pacela, S. Vallecorsafac, M. Spiropulu et al., *Calorimetry with Deep Learning: Particle Classification, Energy Regression, and Simulation for High-Energy Physics*, in *Calorimetry with Deep Learning: Particle Classification, Energy Regression, and Simulation for High-Energy Physics* (2017)

[4] P. Baldi, P. Sadowski, D. Whiteson, Nature Communications **5** (2014)

[5] A. Aurisano, A. Radovic, D. Rocco, A. Himmel, M. Messier, E. Niner, G. Pawloski, F. Psihas, A. Sousa, P. Vahle, Journal of Instrumentation **11**, P09001 (2016)

[6] L. Hertel, L. Li, P. Baldi, J. Bian, *Convolutional Neural Networks for Electron Neutrino and Electron Shower Energy Reconstruction in the NOvA Detectors*, in *Convolutional Neural Networks for Electron Neutrino and Electron Shower Energy Reconstruction in the NOvA Detectors* (2017)

[7] D. Strigl, K. Kofler, S. Podlipnig, *Performance and scalability of GPU-based convolutional neural networks*, in *Proceedings of the 18th Euromicro Conference on Parallel, Distributed and Network-Based Processing, PDP 2010* (2010), pp. 317–324, ISBN 9780769539393

[8] *Google Cloud TPU System Architecture*, https://cloud.google.com/tpu/docs/system-architecture

[9] S.M. Trimberger, Proceedings of the IEEE **103**, 318 (2015)

[10] G. Martin, G. Smith, IEEE Design and Test of Computers **26**, 18 (2009)

[11] J. Duarte, S. Han, P. Harris, S. Jindariani, E. Kreinar, B. Kreis, J. Ngadiuba, M. Pierini, R. Rivera, N. Tran et al., Journal of Instrumentation **13** (2018)

[12] B. Abi, R. Acciarri, M.A. Acero, G. Adamov, D. Adams, M. Adinolfi, Z. Ahmad, J. Ahmed, T. Alion, S. Alonso Monsalve et al. (DUNE), *Neutrino interaction classification with a convolutional neural network in the DUNE far detector* (2020), `http://arxiv.org/abs/2006.15052`

[13] B. Abi, R. Acciarri, M.A. Acero, G. Adamov, D. Adams, M. Adinolfi, Z. Ahmad, J. Ahmed, J. Ahmed, T. Alion et al. (DUNE), *Deep Underground Neutrino Experiment (DUNE), Far Detector Technical Design Report, Volume II DUNE Physics*, in *Far Detector Technical Design Report, Volume II DUNE Physics* (2020), `http://arxiv.org/abs/2002.03005`

[14] K. He, X. Zhang, S. Ren, J. Sun, *Deep residual learning for image recognition*, in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (IEEE Computer Society, 2016), Vol. 2016-December, pp. 770–778, ISBN 9781467388504, ISSN 10636919

[15] A. Radovic, M. Williams, D. Rousseau, M. Kagan, D. Bonacorsi, A. Himmel, A. Aurisano, K. Terao, T. Wongjirad, *Machine learning at the energy and intensity frontiers of particle physics* (2018)