Machine Learning for Sub-Microsecond Edge Data Processing

Audrey C. Therrien[1], *Member, IEEE,* Berthié Gouin-Ferland[1], *Student Member. IEEE,*
Mohammad Mehdi Rahimifar[1], *Student Member. IEEE,* Quentin Wingering[1], *Student Member. IEEE,*
and Ryan Coffee[2].

I. Introduction

As detector technologies improve, the increase in resolution, number of channels and overall size create immense bandwidth challenges for the data acquisition system, extend data center compute times and grow data storage costs. Much of the raw data does not contain useful information and can be significantly reduced with real-time smart veto algorithms, online data compression and online analysis right at the edge [1].

The improvements in artificial intelligence, particularly the many flavours of Machine Learning (ML), adds a powerful and versatile tool to data acquisition (DAQ) strategies [2]. However, large and deep neural networks remain memory and compute intensive, limiting their usability at the edge. One of the most important aspects of integrating ML in a DAQ system is determining when and where integrating a machine learning algorithm will be most beneficial and how to minimize the model size without losing the precision and accuracy required for a scientific application. Furthermore, the performance of the algorithm needs to be measured for both accuracy and compute metrics.

II. Target application
The CookieBox at LCLS-II

New developments in radiation and photonic detectors improve resolution, sensitivity, size and rate, all of which contribute to a gigantic increase in data production rate. One current example of this is the LCLS-II upgrade, which increases the X-ray shot repetition rate from 120 Hz to 1 MHz, in addition to increasing brightness by 4 orders of magnitude [3]. The instrumentation has also undergone improvements, increasing the number of pixels and dynamic range, which contributes to an immense growth in raw data generation [4]. One of the many solutions being implemented to handle this large data rate consists of vetoing bad shots - that is, shots that do not conform to the experimental requirements [5]. To do so, we need to analyze each X-ray pulse shape in time to stop the data collection from downstream detectors.

The CookieBox is an attosecond angular streaking detector used for X-ray pulse shape recovery - that is, it can determine the time and energy spectra of an X-ray shot without significantly altering it [6]. With this information we can veto shots that do not conform to the current experiment requirements and provide additional information on good shots to accelerate the analysis of downstream detector data. However, for this information to be useful, it needs to arrive within a few microseconds to avoid large buffering requirements. Thus, the CookieBox analysis must be completed in less than a microsecond, from the spectrometer analog signal to the control computer, to stay within the 1 MHz rate of pulses.

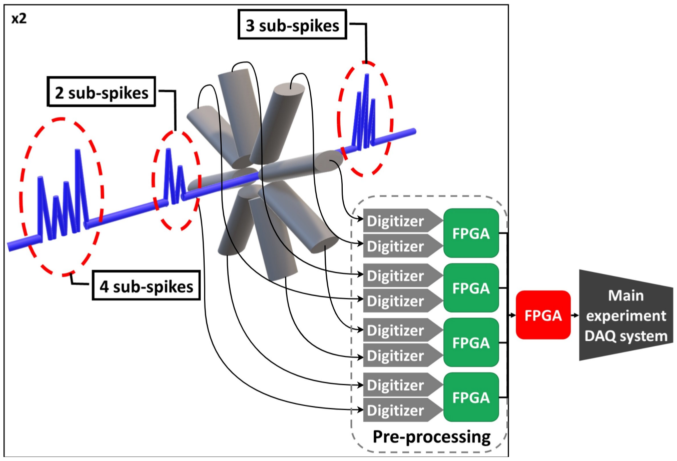


Fig. 1. Overview of the Cookiebox detector. X-ray pulses have different numbers of sub-spikes. The Cookiebox (only 8 channels are shown) samples the pulse with Time-of-Flight Spectrometers. The data is digitized, then preprocessed before converging on a single FPGA which hosts the ML model.

The CookieBox spectrometers generate a total data stream of about 800 GB/s on 16 channels. The data needs to converge on a single node - in this case a single FPGA - for the analysis to take place. For our example system, we developed an algorithm which determines how many sub-spikes exists within the X-ray shot. In other words, we do a simplified time-spectrum analysis to classify a shot as having 0, 1, 2, or many sub-spikes. Most LCLS experiments target single spike or double spike shots, so this real-time analysis can veto shots which do not contribute useful information to the current experiment before they are saved on disk, reducing the size of the dataset.

III. Methodology

A. Data preprocessing

The classical method to obtain the time and energy spectrum of an X-ray shot requires an iterative compute intensive

[1]Interdisciplinary Institute for Technological Innovation, Sherbrooke, Canada.
[2]SLAC National Accelerator Laboratory, Menlo Park, USA.
e-mail: audrey.corbeil.therrien@usherbrooke.ca.

algorithm which cannot be converted to an efficient FPGA implementation [7]. As the initial use of this reconstruction is to identify the number of spikes in a shot, we opted to design a neural network to classify shots. In addition, the combination of signal processing algorithms and compression algorithms with ML can improve the latency and accuracy of edge systems by reducing the width and depth of the model. Thus, we included a peak-finder algorithm and non-uniform quantization of the data before the neural network, which resulted in much smaller neural network models [5].

The information in the spectrometer signal is held in the timestamps of the peaks. Thus, the first operation applied on the signal is a peak-finding algorithm based on first and second order derivatives.

The resolution of the timestamps affects the ability of the downstream neural network to identify the peaks. However, the detector physics imply that not all of the time window used is affected similarly; there are more peaks appearing closely together at the beginning of the window, and the peaks appearing later are fewer and more spread out. We can take advantage of this distribution and optimize a non-uniform width for the bins of the histogram [5]. This minimizes the bit-width necessary for the peak timestamp data, which in turns reduces the size of the neural network and its latency.

B. Neural Network Design

Both Fully Connected Neural Networks (FCNN) and Convolutional Neural Networks (CNN) were trained for performance comparison [5]. They were trained with synthetic data, using the Sparse Categorical Cross-Entropy loss function with the Adam Optimizer and a learning rate of 0.001. Based on previous studies, the FCNN using a width of 5 bits for the optimized non-uniform quantification was selected for this study, as the model that balanced the best accuracy with the lowest computation requirements. The hyperparameters for this model are summarized in Table 1 and Figure 2 shows the confusion matrix.

Input size	32x16
Layers	3xRELU
Output	Softmax
Output size	1x5
Number of parameters	3,433
Accuracy	84 %

TABLE I
NEURAL NETWORK METRICS

C. FPGA implementation

Both the preprocessing steps and the model were implemented on an AMD Xilinx VCU128.

The data preprocessing algorithms, the peak finder and the non-uniform quantizer, were implemented using High Level Syntesis (HLS) C language, and were then imported in the main project as modules.

The neural network model was converted for fast inference on FPGA using the `hls4ml` framework [8]. The hls4ml framework can import standard format model (Keras, ONNX)

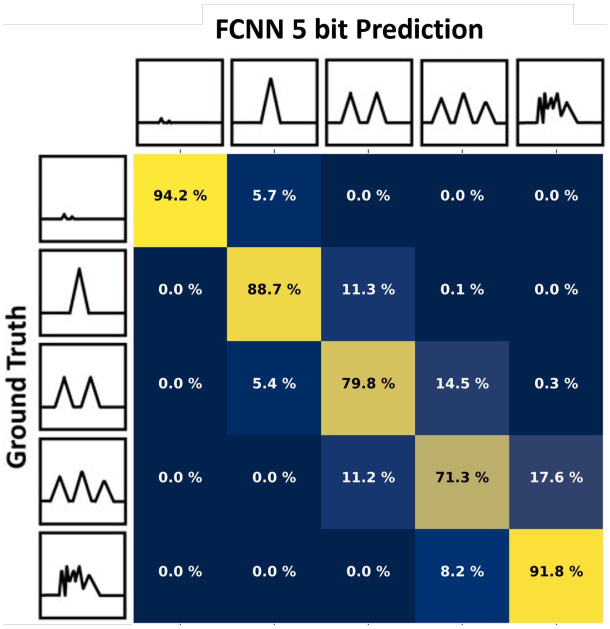


Fig. 2. Confusion matrix for the 5 bit FCNN network

and based on the selected configuration, will generate an IP block that can then be imported in the main FPGA project.

Both the preprocessing modules and the neural network were combined in the project along with VHDL modules for the inbound communication (from the digitizers) and the outbound communication (ethernet port to the computer). For more details on the FPGA architecture, please refer to [9].

D. Testbench

To test the entire data acquisition chain, we built a test bench using a 65 GSPS arbitrary waveform generator (Keysight Technologies M8195A) to replicate the analog output from one of the 16 CookieBox channels [10]. The signal is then digitized by an 8-bit 6.4 GSPS ADC (ADC08DJ3200) before being sent into the FPGA for processing.

Since the current testbench is limited to emulating one channel, we created two modes to test different features. The first one, latency mode, uses the single analog channel with other channels being stored in FPGA memory. This mode is used to test the latency and throughput of the system. The second one, serial mode, transfers the data from each analog channel sequentially, which is then combined in the FPGA. This slower mode lets us measure the impact of the hardware on accuracy.

The processed data is sent to a computer through a 1 Gb/s Ethernet link where the results can be compared to ground truth. An oscilloscope is used to measure the latency between the synchronization signal from the waveform generator and the ”DONE” signal from the FPGA, which signals the end of an inference.

IV. Results and Discussion

A. Implementation

The implementation report including resource utilization and latency of the elements of the data processing chain

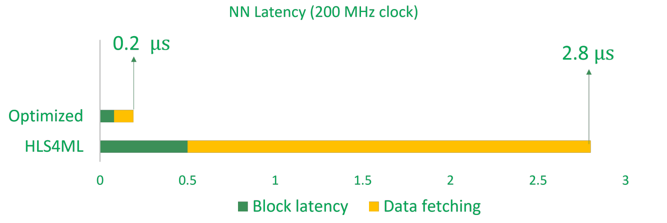


Fig. 3. Latency of the neural network module before and after data fetching optimization.

is summarized in Table 2. The total power average power is 5.28 W. Serial mode and parallel mode have no notable difference in resource usage.

The FCNN originally required 2.8 µs to perform the prediction while the CNN required 50 µs, thus the 5 bit FCNN was selected for further testing. However, to achieve sub microsecond latency, the HLS code generated by hls4ml code was reworked to improve data fetching cycles by increasing the bus width and parallelize the data fetching operation. This improved the FCNN latency to 0.2 µs (figure 3). Adding the latency for the data preprocessing modules, the data processing chain processing predicted time adds up to 0.35 µs. In the next step, we measure the latency of the entire chain using a test bench emulating one channel of the entire acquisition chain.

	Peak-Finder	Quantization	FCNN	CNN
Latency (µs)	0.05	0.1	0.2*	50
DSP	0	0	141	350
LUT	1985	580	14k	85k
BRAM	0	0	4	76
FF	2252	760	7k	31k

*After optimization

TABLE II
VCU128 IMPLEMENTATION AND PERFORMANCE

B. Full chain tests

The testbench consists of, in sequence, the arbitrary waveform generator, the digitizer, the FPGA and the computer. Synthetic data was uploaded to the waveform generator and the results from the data processing chain were sent to the computer where they were compared to the expected results. Since the waveform generator has limited memory, we can only test a small subset of data at a time. The subset used in these experiments had an expected accuracy of 89 %.

On the testbench, the neural network slightly outperformed the expected results of the test dataset with a 90 % accuracy. After verification with our monitoring system, it was found that small differences due to the digitizing process were the cause. Indeed, due to analog noise, the values sent through the waveform generator and then digitized by the ADC varied slightly from the original values. Specifically, some peak values were slightly offset in time compared to the original, leading to a slight difference between the expected and the measured accuracy as shown on figure 4. It should be noted that tests with varying levels of noise and distortion are planned to test the robustness of the DAQ system, something that would not be possible with the CookieBox system and can only be achieved using the experimental testbench.

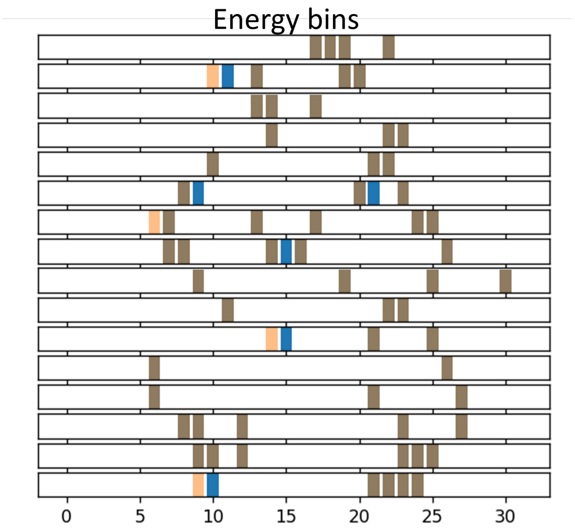
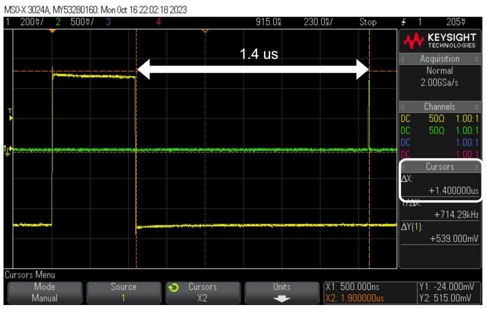


Fig. 4. Differences between the original dataset and the signals after digitization in the DAQ. Ground truth is yellow, testbench digital signal is in blue, and brown indicates both agree on the same value.

The latency was measured between the waveform generator, which outputs a sync signal with the beginning of the 1 µs length analog signal, and the ”DONE” signal which indicates the neural network module has completed an inference. Thus, the time between the falling edge of the waveform generator pulse (in yellow on figure 1) and the ”DONE” signal from the FPGA (green on figure 1) includes the 1 µs analog signal from the CookieBox channel in addition to the entire DAQ processing time. This total time is 1.4 µs, which means the measured processing time is 0.4 µs, less than the time of the signal itself. Thus, this system provides veto information before the next shot and can be used to indicate whether or not data from the downstream detectors needs to be saved with minimal memory buffers.


MSO-X 3024A, MY53280160: Mon Oct 16 22:02:18 2023

Acquisition: Normal 2.00GSa/s

Cursors: ΔX: +1.400000us; 1/ΔX: +714.29kHz; ΔY(1): +539.000mV

Cursors Menu: Mode Manual; Source 1; Cursors X2; Units

X1: 500.000ns; X2: 1.900000us; Y1: -24.000mV; Y2: 515.00mV

Fig. 5. Latency measurement from the waveform generator output to the FPGA output.

V. Conclusion

We developed a system that completes data preprocessing and a neural network inference in 0.4 µs for the CookieBox veto application. The neural network accuracy was slightly affected positively by the variations inherent in the analog to digital conversion process. This effect must be further studied to quantify the robustness of the system to various types of noise. This experiment was enabled by a versatile testbench based on an arbitrary waveform generator that can replicate analog and digital signals from a variety of sensors and can distort them to study DAQ chain robustness to various phenomena.

Artificial Intelligence, in particular machine learning, is a very versatile and powerful tool to analyze and compress data. However, it comes with costs: it requires large training datasets and significant compute power. Thus, its benefits must always be balanced with its requirements and the algorithm complexity should be tailored for a specific function.

Furthermore, the data representation selected for a machine learning algorithm has an immense impact on its performance. Preprocessing the data and choosing a compact, information rich representation helps minimize the size of the machine learning model and thus improve its compute performance significantly.

Finally, it is critical that for science applications the model is validated and tested thoroughly on both valid and invalid data. Systems should have monitoring processes in place to detect when smart DAQ systems operate outside the range they were designed for.

REFERENCES

[1] A. C. Therrien, B. Gouin-Ferland, and M. M. Rahimifar, “Potential of edge machine learning for instrumentation,” *Applied Optics, Vol. 61, Issue 8, pp. 1930-1937*, vol. 61, pp. 1930–1937, 3 2022.

[2] 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*, vol. 13, pp. P07 027–P07 027, 7 2018. [Online]. Available: https://iopscience.iop.org/article/10.1088/1748-0221/13/07/P07027

[3] R. Schoenlein, “New science opportunities enabled by lcls-ii x-ray lasers,” *SLAC Report SLAC-R-1053*, pp. 1 – 189, 2015.

[4] J. B. Thayer, G. Carini, W. Kroeger, C. O’Grady, A. Perazzo, M. Shankar, and M. Weaver, “Building a data system for lcls-ii,” in *2017 IEEE Nuclear Science Symposium and Medical Imaging Conference, NSS/MIC 2017 - Conference Proceedings*. Institute of Electrical and Electronics Engineers Inc., 11 2018.

[5] B. Gouin-Ferland, R. Coffee, and A. C. Therrien, “Data reduction through optimized scalar quantization for more compact neural networks,” *Frontiers in Physics*, vol. 0, p. 887, 9 2022.

[6] N. Hartmann, G. Hartmann, R. Heider, M. S. Wagner, M. Ilchen, J. Buck, A. O. Lindahl, C. Benko, J. Grünert, J. Krzywinski, J. Liu, A. A. Lutman, A. Marinelli, T. Maxwell, A. A. Miahnahri, S. P. Moeller, M. Planas, J. Robinson, A. K. Kazansky, N. M. Kabachnik, J. Viefhaus, T. Feurer, R. Kienberger, R. N. Coffee, and W. Helml, “Attosecond time-energy structure of x-ray free-electron laser pulses,” *Nature Photonics*, vol. 12, pp. 215–220, 2018.

[7] S. Li, Z. Guo, R. N. Coffee, K. Hegazy, Z. Huang, A. Natan, T. Osipov, D. Ray, A. Marinelli, and J. P. Cryan, “Characterizing isolated attosecond pulses with angular streaking,” *Optics Express*, vol. 26, p. 4531, 2 2018.

[8] T. Aarrestad, V. Loncar, N. Ghielmetti, M. Pierini, S. Summers, J. Ngadiuba, C. Petersson, H. Linander, Y. Iiyama, G. D. Guglielmo, J. Duarte, P. Harris, D. Rankin, S. Jindariani, K. Pedro, N. Tran, M. Liu, E. Kreinar, Z. Wu, and D. Hoang, “Fast convolutional neural networks on fpgas with hls4ml,” *Machine Learning: Science and Technology*, vol. 2, p. 045015, 12 2021. [Online]. Available: https://iopscience.iop.org/article/10.1088/2632-2153/ac0ea1

[9] M. M. Rahimifar, Q. Wingering, B. Gouin-Ferland, R. Coffee, and A. C. Therrien, “Accelerating high data rate acquisition: A low-latency FPGA-based EdgeML approach,” *Machine Learning: Science and Technology*, 2024, *under review*.

[10] Q. Wingering, M. M. Rahimifar, and A. C. Therrien, “A versatile edge machine learning test bench for high bandwidth instrumentation,” in *2023 IEEE Nuclear Science Symposium and Medical Imaging Conference*, 2023.