Available on CMS information server **CMS CR -2020/109**

The Compact Muon Solenoid Experiment

Conference Report

Mailing address: CMS CERN, CH-1211 GENEVA 23, Switzerland

30 May 2020 (v3, 13 August 2021)

40 MHz Scouting with Deep Learning in CMS

Dejan Golubovic[1], Thomas Owen James,
Emilio Meschi, Ema Puljak,
Dinyar Rabady, Awais Zahid Rashid,
Hannes Sakulin, Emmanouil Vourliotis[2],
Petr Zejdl[3] on behalf of the CMS Collaboration

Abstract

A 40 MHz scouting system at CMS would provide fast and virtually unlimited statistics for detector diagnostics, alternative luminosity measurements and, in some cases, calibrations, and it has the potential to enable the study of otherwise inaccessible signatures, either too common to fit in the L1 accept budget, or with requirements which are orthogonal to "mainstream" physics, such as long-lived particles. Deep learning is a class of machine learning algorithms that uses multiple layers to progressively extract higher-level features from the raw inputs. A series of studies on different aspects of LHC data processing have demonstrated the potential of deep learning for CERN applications. The usage of deep learning aims at improving physics performance and reducing execution time. This talk will present a deep learning approach to muon scouting in the Level-1 Trigger of the CMS detector. The idea is to utilise multilayered perceptrons to "re-fit"' the Level-1 muon tracks, using fully reconstructed offline tracking parameters as the ground truth for neural network training. The network produces corrected helix parameters (transverse momentum, η and ϕ), with a precision that is greatly improved over the standard Level 1 reconstruction. The network is executed on an FPGA-based PCIe board produced by Micron Technology, the SB-852. It is implemented using the Micron Deep Learning Accelerator inference engine. The methodology for developing deep learning models will be presented, alongside the process of compiling the models for fast inference hardware. The metrics for evaluating performance and the achieved results will be discussed.

Presented at *CTD2020 6th International Workshop Connecting the Dots,*
Published in https://zenodo.org/record/5040675.YRZshtMza3J, doi: 10.5281/zenodo.5040675

[1] Corresponding author

[2] National and Kapodistrian University of Athens, Greece

[3] Also Fermi National Accelerator Laboratory, Illinois, USA

June 29, 2021

40 MHz Scouting with Deep Learning in CMS

Dejan Golubovic[1], Thomas Owen James[1],
Emilio Meschi, Ema Puljak,
Dinyar Rabady, Awais Zahid Rashid,
Hannes Sakulin, Emmanouil Vourliotis[2],
Petr Zejdl[3] on behalf of the CMS Collaboration

CERN, Geneva, Switzerland

ABSTRACT

A 40 MHz scouting system at CMS would provide fast and virtually unlimited statistics for detector diagnostics, alternative luminosity measurements and, in some cases, calibrations, and it has the potential to enable the study of otherwise inaccessible signatures, either too common to fit in the L1 accept budget, or with requirements which are orthogonal to "mainstream" physics, such as long-lived particles. Deep learning is a class of machine learning algorithms that uses multiple layers to progressively extract higher-level features from the raw inputs. A series of studies on different aspects of LHC data processing have demonstrated the potential of deep learning for CERN applications. The usage of deep learning aims at improving physics performance and reducing execution time. This paper will present a deep learning approach to muon scouting in the Level-1 Trigger of the CMS detector. The idea is to utilise multilayered perceptrons to "re-fit" the Level-1 muon tracks, using fully reconstructed offline tracking parameters as the ground truth for neural network training. The network produces corrected helix parameters (transverse momentum, η and ϕ), with a precision that is greatly improved over the standard Level 1 reconstruction. The network is executed on an FPGA-based PCIe board produced by Micron Technology, the SB-852. It is implemented using the Micron Deep Learning Accelerator inference engine. The methodology for developing deep learning models will be presented, alongside the process of compiling the models for fast inference hardware. The metrics for evaluating performance and the achieved results will be discussed.

[1] Corresponding authors
[2] National and Kapodistrian University of Athens, Greece
[3] Also Fermi National Accelerator Laboratory, Illinois, USA

1 Introduction

Collisions in the CERN LHC occur at a bunch crossing (BX) rate of 40 MHz, and generate hundreds of Tb/s of data in the detector electronics. The CMS detector [1] contains $\sim 10^8$ electronic channels, making readout and analysis of every channel at the full BX rate unfeasible. A two-level trigger system selects the events to be read out for permanent storage and subsequent analysis. The Level-1 (L1) trigger [2], implemented in custom hardware using field programmable gate array (FPGA) devices, uses coarse-grained information from the calorimeter and muon subdetectors to search for signatures of interesting physics, and selects events at a maximum rate of 100 kHz. The High Level Trigger (HLT) [2] is a farm of processors analysing full events read out at the L1-accept rate, using complex software algorithms to further reduce the event rate to about 1 kHz to be stored for offline analysis. After the planned Phase-2 upgrade of CMS [3], around 2027, the L1 will include information from the tracking detectors. The L1 and HLT accept rates will be increased to 750 and 7.5 kHz respectively.

Machine learning (ML) describes a set of algorithms that can perform a specific task without being explicitly programmed to do so. Parameters of machine learning models are learned during the *training*, allowing data to shape and influence the algorithm implementation. Deep learning (DL) is a subset of machine learning algorithms that contain multilayered learning systems, such as neural networks (NN) [4]. In principle, any function may be approximated by a NN. Training is performed with the objective of minimizing the model loss, represented by a function of the difference between the targets and the model predictions.

The huge amount of data collected by experiments at the LHC can be used for training machine learning models. With the improvement in computing power and the expansion of DL accelerators, there is a possibility to apply DL techniques in various stages of data acquisition.

2 L1 Scouting

The two-level trigger of CMS is designed to provide excellent sensitivity to most of the interesting physics. Some specific signatures for new physics, however, are either too common to fit in the L1 accept budget, or with requirements which are orthogonal to "mainstream" physics. A "scouting" system using L1 intermediate data at the beam-crossing rate of 40 MHz and carrying out online analyses based on these limited-resolution data may be key for complete coverage of these signatures. For example, in the case of rare Higgs boson decays ($H \rightarrow J/\psi\,\gamma$, $H \rightarrow \phi\gamma$, $H \rightarrow \rho\gamma$), assuming the mass resolution is sufficient, scouting might enable higher signal efficiencies with lower photon thresholds and a broader resonance mass window. For displaced muons, higher efficiency could be obtained by relaxing track-muon matching or combining muons with calorimeter information [5]. Additional examples include flavour anomalies, $B_s \rightarrow \tau\tau$ decays and other low-momentum τ signatures, hadronic physics and QCD measurements [3]. These and other cases will be investigated further, in some cases also profiting from work that can be carried out during LHC Run-3 [3].

The concept of scouting in CMS was initially pioneered at the HLT in 2011 [6]. HLT scouting allows the collection of some reduced-event-content data streams at much higher than the HLT accept rate, but still requires the event to pass the L1 trigger and the complete event data to be read out from the detector through the standard data acquisition chain.

The L1 scouting consists of capturing, reducing, and analysing trigger-level information from the various L1 trigger processors, and storing only relevant high-level information about physics objects. It was first demonstrated in 2018, during the the final weeks of LHC Run-2 [5]. The demonstrator system captured the output of the Global Muon Trigger (GMT), containing the highest-ranking muon candidates as identified by the various hardware track-finders connected to the muon detectors of CMS. About 10^{12} non-empty bunch crossings were collected in the two campaigns, including both proton-proton and heavy-ion runs. Data collected by this prototype system are currently being analysed and preliminary results from one of the analyses indicate that, for example, the muon counts can be used to estimate individual bunch luminosity, with resolution comparable to the other luminosity systems in CMS [3]. This demonstrator will be extended for LHC Run-3.

The CMS Phase-2 L1 trigger upgrade will include an extensive scouting system, capturing intermediate trigger data from the tracking, calorimeter and muon systems, as well as the input and output of the particle flow processors. Capturing the inputs to the Global Trigger (GT) at 40 MHz will enable detailed diagnostics of the trigger system at large. It will be possible to detect anomalies in quasi-real-time in most of the lower level systems by analysing the occupancy and characteristics of the various candidate objects with an effectively unlimited amount of data. It will also be possible to try out novel GT algorithms, as well as cross-check existing ones on a BX-by-BX basis. Additionally, multi-BX correlations will enable detection and analysis of pre-/post-firing, and selection of cosmic ray muons to be used to test L1 tracking efficiency. Although the Phase-2 BRIL [7] system is designed to provide redundant measurements of the luminosity, independent of the L1 trigger, the ability of the scouting system to select and reconstruct specific physics objects or processes without rate limitations or trigger bias, will allow cross-checks of the luminosity measurements and simplify their comparison with those of other experiments.

2.1 Demonstrator System

The scouting demonstrator system contains both FPGA-based and standard software-based processing units. Input data arrives from the GMT over eight 10 Gb/s optical links. The input board (Xilinx KCU1500 [8]) uses a pair of four-lane optical to electrical interfaces to receive the data and a KU-115 Xilinx FPGA to perform data processing. The board decodes the GMT link protocol, aligns the eight links with respect to each other, and performs zero suppression, reducing the data rate by a factor of twenty for the proton-proton collisions [5]. The data are then buffered into FIFOs and sent from the board to the host (Dell R720) via PCIe Gen3 using the Xilinx DMA engine. In software, a fine-grained zero suppression is performed, reducing data rate by another factor of eight. The data are buffered in a RAM disk, and sent to the network over 10 Gb Ethernet. The data then passes through a 10/40 Gb Ethernet switch, before being received by another Dell R720 server with a RAM disk. Here the BZIP [9] algorithm compresses the data by a factor of two. The data are then sent to an eight TB RAID disk for persistent storage. Data are subsequently transferred for processing over an InfiniBand switched network to a distributed file system based on Lustre. About one TB/day of compressed data was recorded during the last few days of proton-proton data taking in LHC Run-2.

For LHC Run-3, starting in 2021, the scouting demonstrator will be extended to include data streams from the Calorimeter Trigger and the Barrel Muon Track Finder (BMTF), requiring four aditional FPGA-based boards. In addition to hardware enhancements, the firmware will be upgraded to include deep learning refitting of the muon tracks, and other corrections applied with machine learning inference.

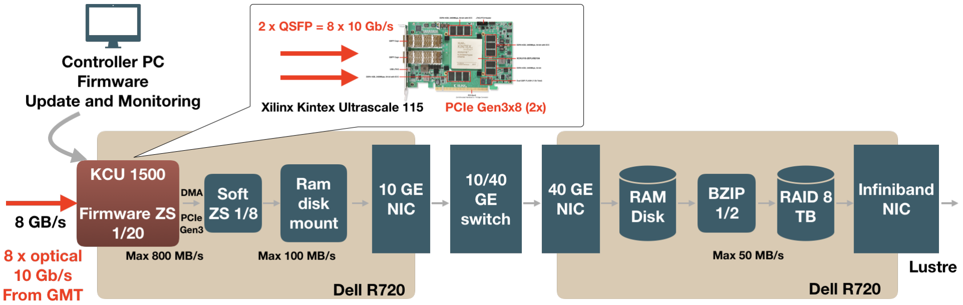

Figure 1: The LHC Run 2 scouting demonstrator system architecture, consisting of two data processing servers, one of which hosts an FPGA-based processing board, the Xilinx KCU1500 [8].

3 Deep Learning for 40 MHz Scouting

The L1 trigger does not have access to the full resolution and granularity data from all CMS sub-detectors, which are used to reconstruct accepted events offline. In order to reach a decision for each bunch-crossing within the available latency of around 3 μs, the L1 muon system uses data from the muon chambers only, and the parameter resolution is optimized to achieve the best efficiency for interesting physics signals, at the minimum total rate. In particular, the p_T measurement in the L1 muon trigger is defined so that when selecting muons with a measured p_T greater than or equal to a certain threshold, the trigger condition is 90% efficient for muons with a true p_T equal to the threshold value. The p_T measurement of the L1 trigger is therefore by definition not an estimate of the true p_T and not suitable to be used directly for a physics analysis. The objective of the following study is to use a DL algorithm to recalibrate the GMT muon parameters for the best resolution, by training the NN using matching offline reconstructed muon tracks as the target. To this end a DL model using as input the GMT muon parameters was trained on real data collected during LHC Run-2, using a special L1 trigger known as ZeroBias. The ZeroBias trigger is a beam bunch crossing-time trigger, without physics signal requirements, used to understand the underlying event structure of collisions occurring at CMS. These triggers are operated with a large prescale, and therefore at a much lower rate than is possible with L1 scouting. ZeroBias datasets (from 2017 and 2018) were used to train the neural networks for L1 scouting so as to ensure an unbiased training sample. In the CMS coordinate system, a muon can be described with a certain pseudorapidity η, transverse momentum p_T with respect to the beam axis, and azimuthal angle ϕ with respect to the counterclockwise beam axis. The L1 muon tracks generated by the L1 Barrel Muon Track Finder (BMTF) (a subsystem of the L1 muon trigger that feeds muon track candidates from the barrel region of CMS to the GMT) are matched to the offline reconstructed muons with a selection of $\Delta R^2 = d\phi^2 + d\eta^2 < 0.01$, as calculated at the second station of the barrel muon detector. Additionally, only L1 muon candidates with $2.5 < p_\mathrm{T} < 45$ GeV are used for training.

3.1 Baseline Model

An L1 trigger muon object is a 64 bit representation of a muon track [10]. Bits are assigned to different muon parameters, including ϕ and η, both at the second muon station, and after extrapolation back to the vertex, the p_T from the track finder, and the muon charge. These parameters can be used as input to the DL model. As the kinematics at the interaction vertex are the most relevant for physics analysis, the following study uses the parameters of the muon extrapolated to the vertex as NN inputs.

The models were implemented using the Keras [11] framework. The baseline model used is a multilayered perceptron with the following characteristics:

- An input layer with four inputs, and an output layer with three outputs. Three hidden layers, with 32 nodes per layer.
- The model inputs are integer values from the L1 trigger muon objects: ϕ, η, p_T, and charge sign. Before training and inference, the inputs are normalised to the range [0-1].
- The prediction targets are three floating point values, representing the difference between the L1 and offline reconstructed ϕ, η and p_T. Before training, the target outputs are also normalised to the range [0-1].
- The activation in the hidden layers is the rectified linear unit $ReLU$ function, and linear in the output layers.
- Batch normalisation (BN) [12] is used before each activation in the hidden layers.
- The learning rate optimizer is *Adam* [13], with the default parameters. The loss function is mean squared error.

3.2 Evaluation Metrics

In order to determine if the NN approach successfully recalibrates the raw GMT values, the distribution of differences between the GMT outputs and the offline reconstructed values is compared with the distribution of differences between the neural network outputs and the offline reconstructed values. Three metrics are used to compare model performance.

- The root mean square (RMS).
- The percentage of data in the distribution core. The distribution core is defined as $\pm 8\,\mathrm{mrad}$, ± 0.0015, and $\pm 1\%$ around zero for $\Delta\phi$, $\Delta\eta$, and $\Delta p_{\mathrm{T}}/p_{\mathrm{T}}$ respectively.
- The percentage of data in the distribution tails. For ϕ, the data are considered to be in the tail if the absolute difference is more than 0.1 radians. For η, the data are considered to be in the tail if the absolute difference is more than 0.05. For p_{T}, the data are considered to be in the tail if the absolute difference is greater than 15% of the reconstructed p_{T}.

3.3 Hyper-parameter Optimization

Hyper-parameters are parameters of the model that are not learned during training, but selected by hand. In this study, an incremental approach is applied to optimize the choice of these parameters. A five-fold cross validation was used with the following handles: the loss function, the optimizer, the regularization, and the hidden and output layer activation functions.

Multiple loss functions were tested: mean squared error, mean squared logarithmic error, and logcosh. A variety of learning rate optimizers were tested: Adam, SGD, Adadelta and Adagrad [14].

Regularization is a technique which prevents models from over-fitting by introducing a small penalty in the training. This also improves training stability. L1 and L2 regularization functions [15] were tested, with parameters (10^{-5}, 10^{-7} and 10^{-9}).

The ReLU activation function was tested in the output layer, but it was unsuccessful in comparison to a linear function. In the hidden layers softmax and ReLU activation functions were tested, with similar results.

Id	Loss function	Optimizer	Regularization	Activation HL	Activation OL
A	logcosh	Adam	none	ReLU	linear
B	logcosh	Adadelta	none	ReLU	linear
C	logcosh	Adadelta	L2 $\lambda=10^{-7}$	ReLU	linear
D	logcosh	Adadelta	L2 $\lambda=10^{-9}$	ReLU	linear
E	logcosh	Adam	L2 $\lambda=10^{-7}$	ReLU	linear
F	logcosh	Adam	none	softmax	linear

Table 1: A description of a some of the most promising models that were tested. The activation functions in the hidden layers (HL) and output layers (OL) are given, alongside the loss functions and optimizer of choice for models labelled A-F.

		RMS			Data in the core [%]			Data in the tail [%]	
Id	$\Delta\phi$	$\Delta\eta$	$\Delta p_{\mathrm{T}}/p_{\mathrm{Tr}}$	$\Delta\phi$	$\Delta\eta$	$\Delta p_{\mathrm{T}}/p_{\mathrm{Tr}}$	$\Delta\phi$	$\Delta\eta$	$\Delta p_{\mathrm{T}}/p_{\mathrm{Tr}}$
GMT	0.166	0.034	0.274	14.82	4.86	10.54	41.25	15.14	44.84
A	0.120	**0.031**	0.169	**20.17**	5.28	14.20	27.07	11.55	28.29
B	0.119	**0.031**	0.167	19.75	5.29	14.76	26.79	**11.45**	26.68
C	0.119	**0.031**	**0.165**	19.37	5.29	**14.85**	26.90	11.58	26.62
D	**0.117**	**0.031**	0.167	19.56	5.27	14.70	**26.73**	11.54	26.84
E	0.118	**0.031**	0.167	20.16	**5.34**	14.61	26.85	11.50	27.04
F	0.119	**0.031**	0.167	19.61	5.24	14.76	26.89	11.54	**26.48**

Table 2: The results of the model configurations described in Table 1. The Δ values refer to the difference between the predicted ϕ, η, or p_{T} and the offline reconstructed values. The Δp_{T} distribution is divided by the offline reconstructed p_{T} (p_{Tr}). The best result for each metric is given in bold.

3.4 Deep Learning Results

The results of the models defined in Table 1 are shown in Table 2. It is observed that each of the models provides an improved estimate of the muon parameters when compared to the raw, uncalibrated GMT outputs. The results show reduced RMS of differences for the three output variables (ϕ, η, p_{T}). The percentage of data in the core is increased, and the amount of data in the tails is reduced, however, the difference between the results of each NN model shown in Tab. 1 is small. No model demonstrates the best performance across all metrics and output variables; for each metric a different set of hyper-parameters produces the best performance. For example, considering the percentage of data in the distribution core, the best performance for ϕ is shown by the model A, for η by the model E and for p_{T} by model C. Similarly, models D, B and F produce the least amount of data in the tails. Model B is used in the following analysis.

Figure 2 shows the distribution of differences between the model predictions and the offline reconstructed measurements. As expected, the neural network outputs produce a narrower distribution around zero than when using the raw GMT values directly.

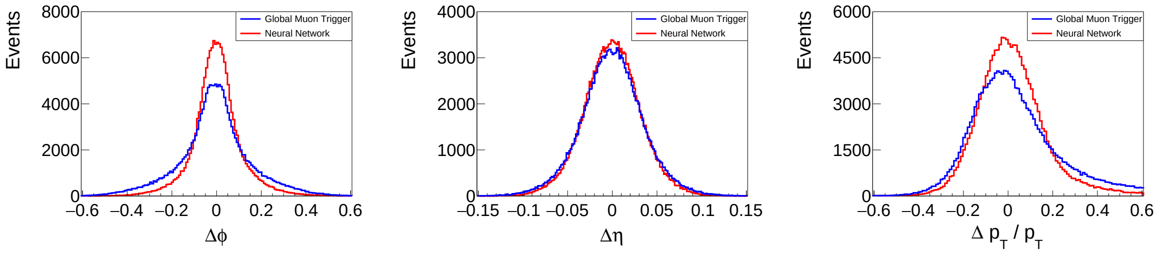

Figure 2: The difference between GMT and offline reconstructed values of ϕ, η, and p_{T} (blue), compared to the difference between the NN predictions and reconstructed values (red). GMT muons with a p_{T} estimate below 3 GeV are excluded because a high fraction of poorly reconstructed muon candidates were found between 2.5 and 3 GeV. While the η coordinate does not require much re-calibration (in the barrel η is almost constant with radius), one can see that the NN is able to estimate the ϕ at the vertex of the track with more precision than the limited granularity of the GMT look-up tables, partially because it can better account for changes in the radius of curvature due to energy losses in lower p_{T} muons. This is because the network learns that for a given ϕ, η, and p_{T} there is a certain average energy loss, which can be accounted for when extrapolating to the vertex.

To evaluate the performance of this DL approach in the context of muon pairs, the MuOnia dataset from 2018 was used. This dataset consists of HLT selected events containing opposite-charge muon pairs and can therefore be used to produce an invariant mass distribution that includes a series of narrow resonances. As shown in Figure 3, the invariant mass distribution in the range $0 < m_{\mathrm{inv}} < 20\,\mathrm{GeV}$ is calculated using three different sets of values: GMT outputs, the NN outputs, and the offline reconstructed values. Vertical lines in this plot correspond to known particles decaying to a muon pair; the ϕ, J/ψ, and Υ meson, with ground state masses of 1.02, 3.09 and 9.46 GeV respectively. The NN demonstrates a better resolution than if the raw GMT values are used. Although the neural network result is unable to distinguish resonances within a particle family, the peaks are narrower and closer to the true resonances than when constructed with raw GMT values; additionally a bump can be observed around the J/ψ meson resonance, which is not present in the raw GMT distribution.

3.5 Deep Learning in FPGA Hardware

The scouting system requires that the DL models be implemented in firmware in an FPGA. The high bandwidth required excludes the use of CPUs or GPUs. Implementing deep learning algorithms using hardware description languages such as VHDL or Verilog is a challenging and time-consuming task. Even though a custom firmware implementation may achieve the best performance in terms of FPGA resource utilisation, it is not as flexible and adaptable as alternative solutions. While CMS is investigating the use

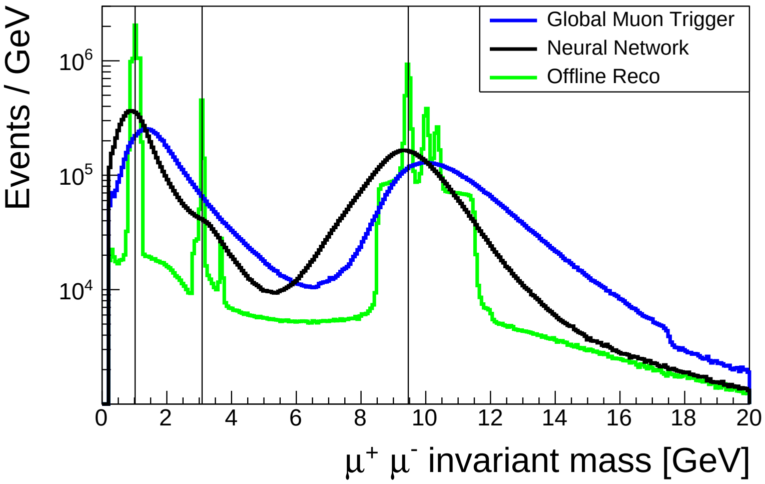

Figure 3: Invariant mass distribution of muon pairs, produced by the CMS MuOnia 2018 dataset. Vertical lines correspond to the ground states of the ϕ, J/ψ, and Υ meson mass resonances.

of high level synthesis languages to create firmware implementations of ML algorithms [16], we instead use a proprietary compiler to translate Python models into instructions for FPGA-based hardware, the Micron Deep Learning Accelerator (MDLA) [17].

The hardware used for the implementation of the developed deep learning models are the Micron SB-852 and Micron AC-510 FPGA-based processing boards [18, 19]. The SB-852 contains the Xilinx Virtex Ultrascale+ VU9P FPGA [20], 64GB DDR4 memory, two pairs of four-lane optical link interfaces and a PCIe x16 Gen3 connector to the host. The board is installed in a rack mounted dual-CPU server for testing purposes. The MDLA allows access to the board from the host using a Python API, providing an easy way to evaluate a variety of deep learning models on the FPGA-based board. The implementation is hidden from the user, eliminating any need to write low level code or firmware. The development process starts with model training in one of the software frameworks such as Keras [11] or TensorFlow [21]. Then, the models are converted to the Open Neural Network Exchange (ONNX) [22] format. After the conversion, the MDLA API is used to communicate with the board, in order to set up the model execution in the FPGA[23, 24]. During development, measurements of latency and throughput are made to evaluate inference performance.

In order to keep up with the incoming data, a throughput of 10^6 inferences per second is required per L1 scouting board. As shown in Table 3, the performance achieved using the MDLA satisfies these requirements. Using two computational clusters instantiated within a single AC-510 board, the requirements are exceeded by almost a factor of three. With the expected future upgrades of the MDLA hardware and software, an increase in performance is expected. The results obtained should allow for the inclusion of the SB-852 hardware in the scouting system, potentially replacing the KCU1500 board.

Hardware	Clusters	Latency [ns]	Inferences per second
AC-510	1	726	> 1 300 400
AC-510	2	364	> 2 700 000
SB-852	1	748	> 1 300 000

Table 3: Scouting neural network (Model B) performance on the MDLA, with the AC-510 and SB-852 FPGA-based boards.

4 Summary and Outlook

Recalibrating L1 muon objects using deep learning models shows potential for further use within the CMS L1 scouting system. It has been established that the use of deep learning models can achieve a re-calibration of L1 muon parameters to make them suitable for determining the invariant mass of pairs of muons. The deep learning models are implemented in FPGA-based hardware, and the measured throughput and latency fit into the CMS scouting requirements. The MDLA will now be integrated into the scouting system for LHC Run-3 and in the future new models will be developed to take advantage of additional inputs to the L1 scouting system.

ACKNOWLEDGEMENTS

We thank Micron Technology Inc. for the financial contribution and the technical support throughout the project.

References

[1] CMS Collaboration, "The CMS experiment at the CERN LHC" JINST **3** S08004 (2008), doi:10.1088/1748-0221/3/08/s08004.

[2] CMS Collaboration, "The CMS trigger system", JINST **12**, P01020 (2017), doi:10.1088/1748-0221/12/01/P01020.

[3] CMS Collaboration, "The Phase-2 upgrade of the CMS Level-1 trigger", CERN-LHCC-2020-004;CMS-TDR-021 (2020).

[4] Y. Bengio *et al.*, "Deep learning", Nature **521**, 436 (2015), doi:10.1038/nature14539.

[5] H. Sakulin *et al.*, "40 MHz Level-1 trigger scouting for the Compact Muon Solenoid experiment", Proceedings of 24th International Conference on Computing in High-Energy and Nuclear Physics, Adelaide, Australia (2020). To be published.

[6] V. Khachatryan *et al.* [CMS], "Search for narrow resonances in dijet final states at $\sqrt(s) =$ 8 TeV with the novel CMS technique of data scouting", Phys. Rev. Lett. **117**, 031802 (2016) doi:10.1103/PhysRevLett.117.031802.

[7] CMS Collaboration, "The Phase-2 upgrade of the CMS beam radiation, instrumentation, and luminosity detectors: conceptual design", CMS-NOTE-2019-008;CERN-CMS-NOTE-2019-008 (2020).

[8] Xilinx Inc., `www.xilinx.com/products/boards-and-kits/dk-u1-kcu1500-g.html`, [Accessed 2020-02-26].

[9] J. Seward, "bzip2", `www.sourceware.org/bzip2`, [Accessed 2020-02-26].

[10] CMS Collaboration, "Scales for inputs to μGT (ϕ, η, $p_{\mathrm{t}}/E_{\mathrm{t}}$), and others", `http://globaltrigger.hephy.at/files/upgrade/ugt/scales_inputs_2_ugt_2017Aug14.pdf`, [Accessed 2020-03-09].

[11] "Keras", `www.keras.io`, [Accessed 2020-03-09].

[12] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift", [arXiv:1502.03167 [cs.LG]].

[13] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization", [arXiv:1412.6980 [cs.LG]].

[14] S. Ruder, “An overview of gradient descent optimization algorithms”, [arXiv:1609.04747 [cs.LG]].

[15] A. Y. Ng, “Feature selection, L1 vs. L2 regularization, and rotational invariance”, International Conference on Machine Learning (ICML) (2004), doi:10.1145/1015330.1015435.

[16] J. Duarte et al., “Fast inference of deep neural networks in FPGAs for particle physics”, JINST **13** P07027 (2018), arXiv:1804.06913.

[17] Micron Technology, Inc., “Micron Deep Learning Accelerator Software Development Kit”, `www.github.com/FWDNXT/SDK`, [Accessed 2020-03-09].

[18] Micron Technology, Inc., `www.micron.com/products/advanced-solutions/advanced-computing-solutions/hpc-single-board-accelerators/sb-852`, [Accessed 2020-02-26].

[19] Micron Technology, Inc., `www.micron.com/products/advanced-solutions/advanced-computing-solutions/ac-series-hpc-modules/ac-510`, [Accessed 2020-03-09].

[20] Xilinx Inc., `www.xilinx.com/products/silicon-devices/fpga/virtex-ultrascale-plus.html`, [Accessed 2020-03-09].

[21] “Tensorflow”, `www.tensorflow.org`, [Accessed 2020-03-09].

[22] “Open Neural Network Exchange format (ONNX)”, `www.onnx.ai`, [Accessed 2020-03-02].

[23] V. Gokhale *et al.*, “Snowflake: a model agnostic accelerator for deep convolutional neural networks”, (2017), arXiv:1708.02579.

[24] A. X. M. Chang *et al.*, “Deep neural networks compiler for a trace-based accelerator”, Journal of Systems Architecture **102**, 101659 (2020), doi:10.1016/j.sysarc.2019.101659.