Meeting the latency and energy constraints on timing-critical edge-AI systems

Ivan Miro Panades, Inna Kucher, Vincent Lorrain, Alexandre Valentian

Meeting the latency and energy constraints on timing-critical edge-AI systems

Ivan Miro-Panades, Inna Kucher, Vincent Lorrain, Alexandre Valentian

***Abstract*— Smart devices, with AI capabilities, at the edge have demonstrated impressive application results. The current trend in video/image analysis is to increase the resolution and classification accuracy. Moreover, computing object detection and classification tasks at the edge require both low latency and high-energy efficiency for these new devices. In this paper, we will explore a novel architectural approach to overcome such limitations by using the attention mechanism of the human brain. The latter allows humans to selectively analyze a scene allowing limiting the spent energy.**

***Index Terms*—Edge AI accelerator, high-energy efficiency, low-latency, object detection.**

I. Introduction

The observed trend, in visual processing tasks, is to increase the complexity of neural network (NN) topologies to improve the classification accuracy. This results in NN models being deeper and larger leading to several issues when used in edge applications. Even though mobile versions of some network topologies have been introduced over time, it remains difficult to integrate them on-chip in an energy-efficient manner. The main issue is the large number of parameters, requiring the use of an external memory which leads to a large power dissipation due to the data movement. It should be noted that the energy necessary for moving data is three orders of magnitude larger than that for doing computations on the same data 0. This is the primary issue ("Issue No. 1") that must be addressed.

Moreover, when processing a video input stream, the whole image is processed, frame by frame, even though there is enormous spatial redundancy between consecutive frames. If the target application requires a low reaction time to events (e.g., to an object or person moving), the frame rate needs to be high, leading to high instantaneous power values. On the other hand, if the frame rate can be kept low, such as for security surveillance applications, the system overall energy efficiency would still be poor, because of the high inter-frame redundancy (especially during nights and weekends). The second issue ("Issue No. 2") that must be addressed is to reconcile low power dissipation and short reaction times to events.

In those respects, bio-inspired approaches can lead to innovative solutions. For instance, neuroscientists have found anatomical evidence that there are two separate cortical pathways, or 'streams', in the visual cortex of monkeys 0: the ventral pathway and the dorsal pathway, as shown in Fig. 1. The dorsal stream is relatively fast, sensitive to high-temporal frequencies (i.e. motion), viewer-centered and relatively unconscious. It has been called the "Where" path, since it is used for quickly retrieving the location of objects, especially of moving ones. On the other hand, the ventral stream is relatively slow (≈4x higher reaction time), sensitive to high-spatial frequencies (i.e., details), object-centered and relatively conscious. It is known as the "What" path, involved in the recognition of objects. If millions of years of evolution has lead to such a 2-path solution, this is because it brought a competitive advantage to our ancestors, allowing them to quickly evade threats, even before their brain was knew the nature of the threat.

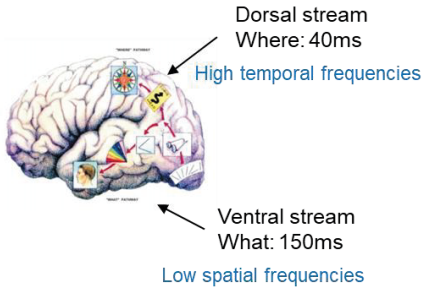


Fig. 1. Illustration of the two visual pathways or streams in the visual cortex, used for extracting different information

Even though this 2-stream hypothesis has been disputed over the years and we now know that those pathways are not strictly independent, but do interact with each other (for instance for grasping fine objects 0), it is still relevant to the problem at hand, as it provides a good fit to many motor and perceptual findings.

In this work, we focus on the "Where" subsystem, as the "What" one is already well addressed with existing accelerators [10]. The main objectives are therefore to obtain the lowest possible latency and power values. First, we have started by selecting an adequate neural network topology, i.e., one with a

This work was in part funded by the ECSEL Joint Undertaking (JU) under grant agreement No 876925. The JU receives support from the European Union's Horizon 2020 research and innovation program and France, Belgium, Germany, Netherlands, Portugal, Spain, Switzerland.

Authors Ivan Miro-Panades and Alexandre Valentian are with Univ. Grenoble Alpes, CEA, List, F-38000 Grenoble, France (e-mail: ivan.miro-panades@cea.fr and alexandre.valentian@cea.fr).

Authors Inna Kucher and Vincent Lorrain are with Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France (e-mail: inna.kucher@cea.fr and vincent.lorrain@cea.fr)

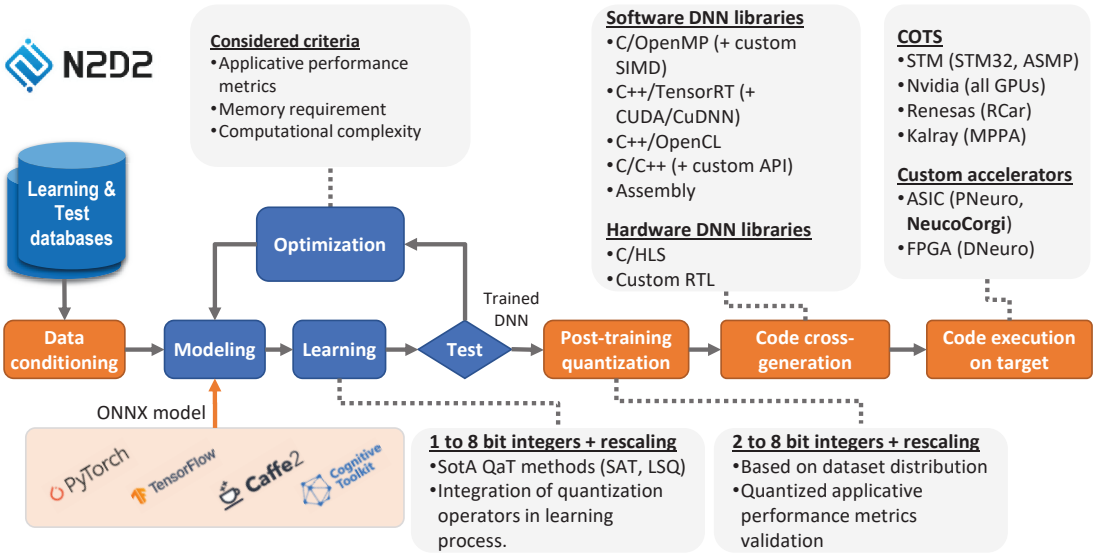


Fig. 2. N2D2 framework

small number of parameters, but with only slightly degraded accuracy: we have chosen the MobileNet-V1 topology 0. Since all the parameters need to be stored on-chip, for solving 'Issue N° 1', the synaptic weights and activation values must be heavily quantized, without a significant loss in accuracy: our in-house learning framework was complemented with a state-of-the-art quantization-aware training (QAT) algorithm. This tool is now available in open source and presented in Section II. An innovative architecture was considered for the hardware, once again taking inspiration from biology: layers V1 to V3 of the visual cortex, which are sensitive to orientations of edges and to movement, are fixed early on during life. For instance, V1 undergoes synaptic and dendritic refinement to reach adult appearance at around 2 years of age 0. Even though these synaptic weights will not be learnt again during adulthood, that does not prevent our visual cortex to learn how to recognize new objects. We have thus chosen to fix the feature extraction layers of the MobileNet once and for all (while ensuring they remain sufficiently generic) and then to apply a transfer learning technique, to target several applications. Fixing synaptic weights actually leads to tremendous energy and latency savings, e.g. getting rid of memory accesses. Such an architecture can be used in an attention mechanism, solving 'Issue N° 2'. The analysis of the architecture is described in Section III. Finally, Section IV concludes this work.

II. Quantization Aware Training

A scalable and efficient QAT tool has been developed and integrated into the N2D2 framework 0 (see Fig. 2). N2D2 provides a complete design environment for a wide range of quantization modes to achieve the best performances including SAT 0 and LSQ 0 methods. The overall framework and the addition of the quantization aware training modules are shown in Figs 3 and 4 below.

The advantages of this dedicated framework include:

- Integration of common quantization methods (Dorefa, PACT, CG-PACT).
- Straightforward support of mixed-precision operators (2-bit to 8-bit for Weights and/or Activations).
- Automatic support of non-quantized layers (e.g., batch normalization).
- Training phase based on optimized computing kernels, resulting in fast evaluation of the quantization performance.

There are two separate quantization modules, one dedicated to weights and another one to activations. This is illustrated in Figs 3 and 4 below, where the example Layer N consists of a Convolutional layer followed by the Batch normalization layer with an activation function (typically ReLu). The weights of this Convolutional layer are quantized to a desired precision, using the quantize_wts function. Batch normalization stays in full precision and goes through the activation function. This output is then quantized to the required precision, using the quantize_acts function. It must be noted that the two quantification precisions, i.e., of the weights and activations, might not necessarily be the same.

During neural network training, the parameters are adjusted using backpropagation of errors on these parameters, and repeating this process for a certain number of epochs, until the figure of merit of the training is satisfactory.

The forward pass with QAT follows the logic shown in Fig. 3:

- Inputs arriving to the convolutional layer are passed through the convolution operation, where the weights are quantized beforehand using Weight Quantizer module;

- The output is propagated to Batch normalization layer, which provides operations in full precision;
- The output from Batch normalization is transformed to its quantized values using the Activation Quantizer;
- At the end, the quantized output is passed as an input to the next layer.

The backward propagation with QAT, shown in Fig. 4, includes the following steps:

- Starting from the errors on quantized activations, the errors on full precision activations are computed, using the derivatives of the transformations applied in the forward pass;
- Then these errors, on full precision activations, are propagated through Batch normalization and convolutional layers;
- In a similar way, the errors on full precision weights are computed using quantized weights errors.

During the learning procedure, both full precision and quantized quantities are kept. One has to keep in mind that, during the training, the applied procedure is called "fake" quantization, since even quantized values are kept using floating-point type.

Once the network is trained, the weights and inputs are transformed into true integer values before execution on a hardware target.

III. Architecture Exploration

The architecture exploration started with the choice of the NN topology, with low energy and low latency per inference in mind: the target is an energy below 4mJ per image (HD: 1280x720 pixels) and a latency compatible with a 30FPS frame rate (i.e. below 30ms). A tradeoff must thus be made between network complexity and operations per inference. A lower number of operations obviously leads to a lower number of Multiplication-Accumulation (MAC) operations to be performed per image.

Fig. 5 illustrates the various topologies that can be found in the literature [9]. The MobileNet-V1 topology has been chosen, as it uses depth-wise and point-wise convolutions to reduce the computing complexity (their difference with a standard convolution is shown in Fig. 6).

Usually, NN accelerators use a layer-wise architecture. This makes it possible to support different topologies, since networks are computed layer-wise. It also makes it possible to compute multiple images per layer, i.e., inputs with a batch size higher than one: the synaptic weights are read once and can be reused for the different images, reducing the power dissipation. However, in our case, we have conflicting constraints: the topology is fixed and the batch size is equal to one to limit the processing latency. A streaming architecture is thus considered,

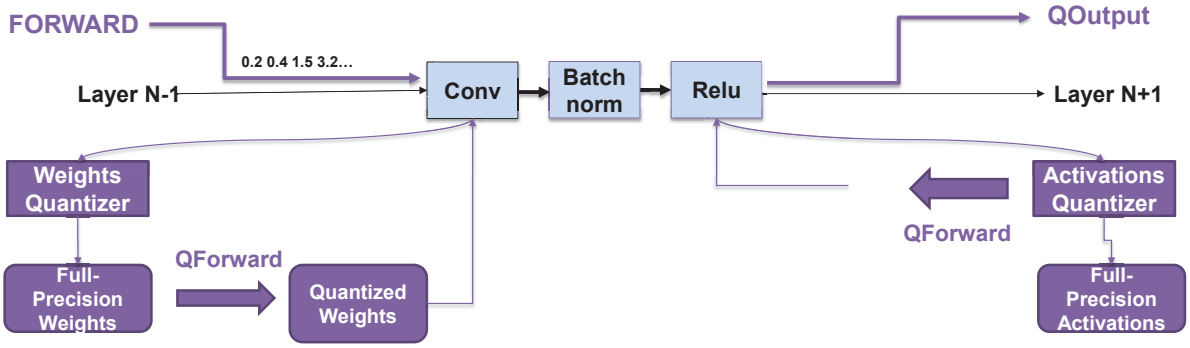


Fig. 3. Forward and backward quantization passes

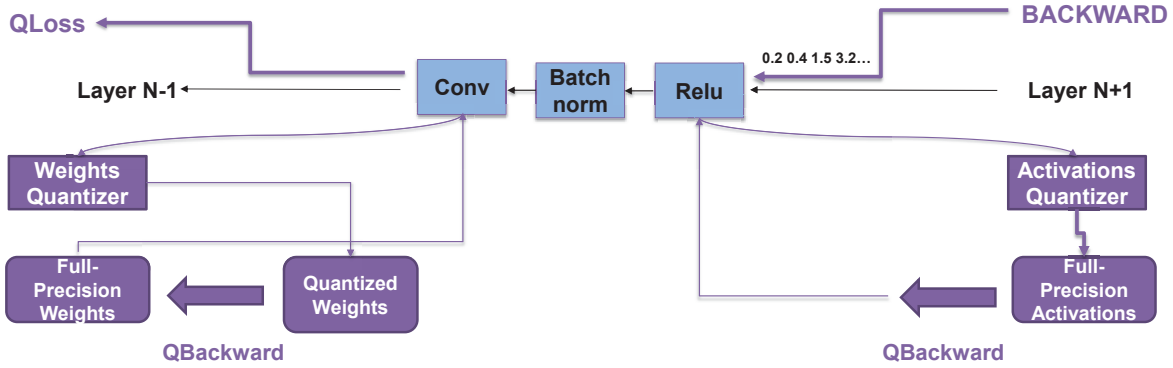


Fig. 4. Backward propagation with QAT

since the latency is minimized and the fixed topology allows optimizing the buffering and the inter-layer communication throughput, limiting the area overhead.

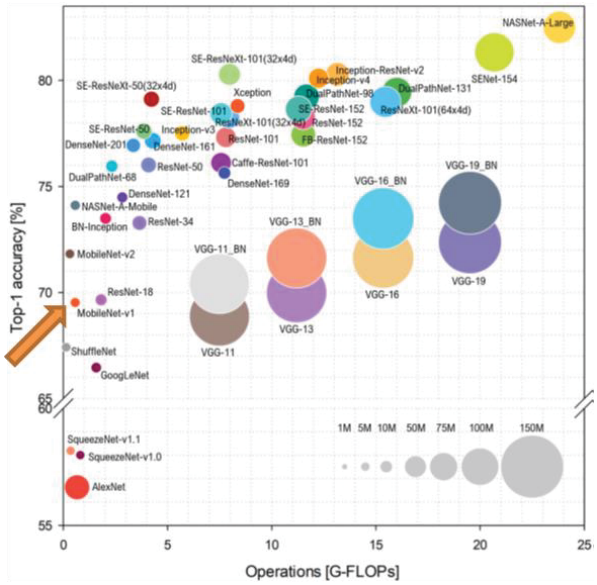
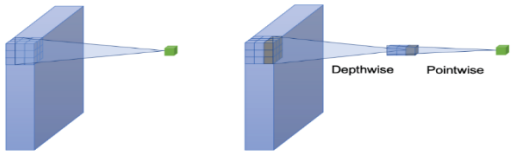


Fig. 5. Comparison of several NN topologies, as function of number of operations (X-axis), classification accuracy (Y-axis) and number of parameters (size of the circle) 0



(a) (b)

Fig. 6. (a) Standard convolution; (b) Depth-wise + point-wise convolution

Our NN accelerator, called NeuroCorgi, thus takes the form of a pipelined computational architecture, in which each layer of the network is instantiated into a specialized, parameterizable sub-architecture. These sub-architectures are then connected according to the network topology and parameterized to perform the inference calculation (conv, FC …) and minimize the latency.

To simplify the architectural tradeoff analysis and the RTL generation, a back-end tool has been added to the N2D2 learning framework. This tool takes as input an algorithmic configuration file (representing the computation that need to be performed per layer) and the hardware parameters for each layer sub-architecture. It then generates files, following a 3 step procedure: first, the generation of the topological and hardware configuration; second, the generation of the RTL code; and finally, the test and validation files.

This tool suite is very useful for architecture exploration; by varying several architectural parameters: level of parallelism of each sub-architecture; size of the buffers between layers, to balance the data flow and minimize the congestion in the pipeline. An exploration of the design space was done by manually varying these parameters: their impact can be readily assessed at accelerator-level. The pipelined architecture allows ultra-low latency image detection (11ms). The result of initial floorplanning experiments is shown in Fig. 7.

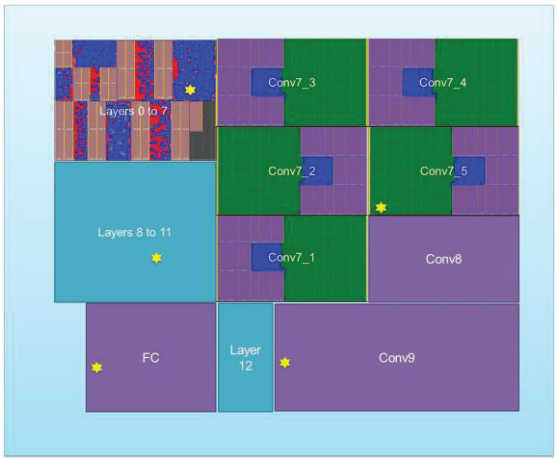


Fig. 7. NeuroCorgi initial floorplan, illustrating the placement of the different NN layers

IV. Conclusions

We aim at solving the paradox of handling ever larger image resolutions (HD) and frame rates (>30FPS), with more complex neural networks, while at the same exhibiting low latency and power values. In this work, we explored a clever, bio-inspired solution, for providing an attention mechanism to vision solutions at the edge. We focus on the dorsal stream, or "Where" path, since the "What" path is already well covered by a number of accelerators.

For pushing the energy efficiency to its maximum, several design decisions were made: a small NN topology was chosen, i.e., MobileNet-V1 to be completely integrable on-chip; weights and activations were heavily quantized (4b); bio-inspiration was again considered, by fixing the features extraction layers (embedded memory limited to 600kB).

Our in-house learning framework N2D2 has been completed with the necessary functionalities: state-of-the-art quantization algorithms, transfer learning, hardware generation and configuration.

References

[1] B. Dally, "CPU Computing To ExaScale and Beyond", *The International Conference for High Performance Computing, Networking, Storage, and Analysis (Super Computing)*, 2010.

[2] M. Mishkin, L.G. Ungerleider and K.A. Macko, "Object vision and spatial vision: two cortical pathways," *Trends Neuroscience*, Vol. 6, pp.414–417, 1983.

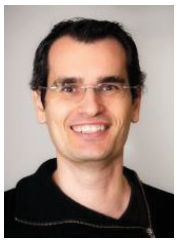
[3] V. Van Polanen and M. Davare, "Interaction between dorsal and ventral streams for controlling skilled grasp," *Neuropsychologia*, 79(Pt B), pp. 186–191, 2015.

[4] A. G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," https://doi.org/10.48550/arXiv.1704.04861.

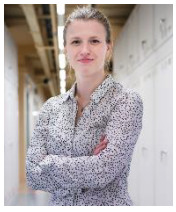
[5] C. R. Siu and K. M. Murphy, "The development of human visual cortex and clinical implications," *Eye and Brain* 2018:10 25–36, doi: 10.2147/EB.S130893.

[6] https://github.com/CEA-LIST/N2D2.

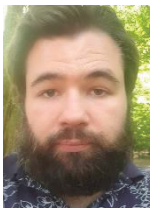
[7] Q. Jin, L. Yang, A. Liao, "Towards Efficient Training for Neural Network Quantization," arXiv:1912.10207 [cs.CV], 2019.
[8] S. K. Esser *et al.*, "Learned Step Size Quantization," arXiv:1902.08153 [cs.LG], 2019.
[9] S. Bianco, R. Cadene, L. Celona and P. Napoletano, "Benchmark Analysis of Representative Deep Neural Network Architectures," in IEEE Access, vol. 6, pp. 64270-64277, 2018, doi: 10.1109/ACCESS.2018.2877890.
[10] A. Reuther *et al.*, "Survey and Benchmarking of Machine Learning Accelerators," *IEEE Conference on High Performance Extreme Computing (HPEC)*, 2019, https://doi.org/10.48550/arXiv.1908.11348.



Ivan Miro-Panades received the M.S. degree in telecommunication engineering from the Technical University of Catalonia (UPC), Barcelona, Spain, in 2002, and the M.S. and Ph.D. degrees in computer science from Pierre and Marie Curie University (UPMC), Paris, France, in 2004 and 2008, respectively. He worked at Philips Research, Sureness, France and STMicroelectronics, Crolles, France, before joining CEA, Grenoble, France, in 2008, where he is currently an Expert Research Engineer in digital integrated circuits. His main research interests are artificial intelligence, the Internet of Things, low-power architectures, energy-efficient systems, and Fmax/Vmin tracking methodologies.



Inna Kucher received the M.S. degree in high-energy physics from École Polytechnique, Palaiseau, France, in 2013, and Ph.D. degree in high-energy particle physics from Paris-Saclay University, Orsay, France, in 2017. She worked in École Polytechnique and Cern, before joining CEA LIST, in 2020, where she is currently a Research Engineer in neural networks development, optimization and deployment on embedded platforms.



Vincent Lorrain received his engineering degree from ESEO, Angers, France, in 2014, a M.S. degree in microelectronics from INSA, Renne, France, in 2014, and his PhD degree in physics from Université Paris-Saclay, Orsay, France, in 2018. He has been working at CEA LIST, since 2018, where he is currently a research engineer in the development of optimized neural network architecture.

Alexandre Valentian joined CEA LETI in 2005, after an MSc and a PhD in microelectronics. His past research activities included design technology co-optimization, promoting the FDSOI technology (notably through his participation in the SOI Academy), 2.5D/3D integration technologies and non-volatile memory technology. He is currently pursuing the development of bio-inspired circuits for AI, combining memory technology, information encoding and dedicated learning methods. Since 2020, he heads the Systems-on-Chip and Advanced Technologies (LSTA) laboratory at CEA LIST.
Dr Valentian has authored or co-authored 80 conference and journal papers.