Find more on our web: hipeac.net/jobs/10893

Inria Rennes
Rennes, France

Efficient low-precision training for deep learning accelerators

- **Deadline**: July 31, 2019
- **Career levels**: PhD student
- **Keywords**: Approximate Computing, Computer Architecture, Energy Efficiency / Low-power Computing, Machine Learning / AI, Reconfigurable Computing

Deep Learning is one of the most intensively and widely used predictive models in the field of Machine Learning. Convolutional Neural Networks (CNNs) [2] have shown to achieve state-of-the-art accuracy in computer vision [1] and have even surpassed the error rate of the human visual cortex. These neural network techniques have quickly spread beyond computer vision to other domains. For instance, deep CNNs have revolutionised tasks such as face recognition, object detection, and medical image processing. Recurrent neural networks (RNNs) achieve state-of-the-art results in speech recognition and natural language translation [3], while ensembles of neural networks already offer superior predictions in financial portfolio management, playing complex games [4] and self-driving cars [5].

Despite the benefits that DL brings to the table, there are still important challenges that remain to be addressed if the computational workloads associated with NNs are to be deployed on embedded edge devices that require improved energy efficiency. For instance, the amazing performance of AlphaGo [4] required 4 to 6 weeks of training executed on 2000 CPUs and 250 GPUs for a total of about 600kW of power consumption (while the human brain of a Go player requires about 20W). Such taxing demands are pushing both industry and academia to concentrate on designing custom platforms for DL algorithms that target improved performance and/or energy efficiency.

One general way to increase the performance and efficiency in computing is through reducing the numerical precision of basic arithmetic operations. In the case of DL systems, there are two main computational tasks: training and inference. Training requires vast quantities of labeled data that are used to optimize the network for the task at hand, usually by way of some form of stochastic gradient descent (SGD) algorithm. Inference, on the other hand, is the actual application of the trained network, which can be replicated onto millions of devices. Between the two, reducing numerical precision during inference has received the most attention from the research community over the last years, with some promising results in certain applications [6,7]. Much less has been done for the training phase, the main reason being that the effects of low-precision arithmetic on training algorithms are not yet well understood. This has by no means stopped major players in the hardware space to start devising architectures that offer increasing support for low-precision arithmetic. In the particular case of training, there are already commercial platforms that mix 32-bit high precision floating-point computing with low precision 16-bit formats for increased performance [8–10].

With this project, we want to conduct a thorough analysis of reduced numerical precision training of DL systems. A first main task is for the PhD student to build/augment a deep learning platform with custom precision arithmetic. This will require building customized floating-point operators down to very few bits of exponent and mantissa which offer a desirable balance between accuracy and energy efficiency.

In parallel, the plan is to investigate how mixed precision support (i.e., hardware support for several numeric formats with varying costs and accuracies) during successive iterations of the SGD training algorithm impacts accuracy and performance. There are two concrete architecture scenarios we have in mind:

- A heterogenous host-accelerator model of computation in which a fast accelerator (e.g. FPGA or ASIC) with support for low-precision arithmetic is connected to a slower general purpose host device (e.g. CPU, GPU) that

The HiPEAC project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement number 779656.

- can perform high-precision arithmetic by way of an infrequently used communication channel. The bit centering idea recently presented in [11] offers one example of this high and low precision combination for SGD-based algorithms in DL contexts.
- An accelerator-only solution specialized on low-precision, but which can be used to implement higher precision operations by way of floating-point expansions [12, Ch. 14]. As the training algorithm converges, higher accuracy is needed for certain operations, which can be implemented using expansions. We plan to investigate the various tradeoffs in computation time, operator complexity and energy efficiency of such an architecture. We will pursue low-precision training tasks for classical architectures using dense weight matrices, but also on compressed network architectures based on structured weight matrices such as block-circulant matrices [13]. These architectures offer the promise of small storage requirements (i.e., linear in the number of neurons of each layer vs quadratic in classic architectures) and improved performance in training and inference (i.e., quasi-linear vs quadratic cost in the classic setting), critical elements needed for a wide adoption of deep learning methods on low-power embedded devices.

With respect to existing works that generally consider predefined numeric formats, our aim is to do a more in-depth analytical design space exploration by looking at the entire spectrum of low precision floating-point arithmetic formats and how the working precision can be effectively varied in-between training iterations.

SGD-based algorithms are derived from first order optimization methods. In the DL world, their simplicity and practical effectiveness for high dimensional problems have made them ubiquitous. Nevertheless, at least from a theoretical perspective, second-order methods (e.g. Newton-based iterations) are very attractive due to their better convergence rates. In practical DL scenarios, second-order methods have not found much use due to a perceived idea that they do not generalize as well as SGD. Still, as recent work shows, this is not always necessarily the case [14]. Based on this, it might also prove worthwhile to explore the use of mixed-precision training for second-order DL methods.

The student will also have the task of validating the developed techniques through a prototype of an accelerator for CNN training. Synthesis of the specialized architecture on a target hardware platform will demonstrate the gains in performance and energy of the automatically generated accelerators. This parallel architecture will include configurable arithmetic operators implementing various precision and number representations as defined by the proposed exploration methodology. Concretely, the architecture will be validated in an FPGA accelerator for CNNs. Second, an ASIC prototype will be designed in a 28nm technology to be able to reach the highest energy efficiency. The team has such experience in designing custom chips, evaluating performance and power consumption in advanced technology, and even going down to a silicon prototype (even if hardly reachable in the frame of a PhD thesis). Comparison with other platforms such as embedded GPUs, and existing FPGA implementation of CNNs will be performed. However, our objective is not to compete with main players such as Nvidia or Google. Instead, our main focus is on energy-efficient embedded systems, such as autonomous vehicles, or on ultra-low-power IoT (Internet of Things) devices.

Host Team: Cairn@InriaThe CAIRN team from Inria (the French national Research Institute for digital sciences) has a long history of working on energy-efficient computing kernels. This project, through its target of energy-efficient DL training, enriches the set of applications being worked on by the team, while also leveraging on our experience working with custom numeric fixed-point and floating-point formats. In particular, we have already demonstrated the potential benefit of small floating-point formats for machine learning applications [15] and we consider them to also be applicable in complex DL systems as well. To this end, in the context of an M2 internship that has just started in the team, we are working on integrating ctfloat, the custom low precision floating-point library for high-level synthesis we have developed, to the N2D2 DL framework developed by CEA LIST, with the goal of constructing energy efficient inference kernels. As part of this PhD thesis proposal, the goal is to pursue and expand this work further for the more complicated and expensive training tasks.

This PhD thesis is funded by the AdequateDL project, sponsored by ANR, the funding agency for research in France.

References[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in Advances in Neural Information Processing Systems 25 (NIPS 2012), 2012, pp. 1097–1105.

[2] Y. LeCun, Y. Bengio, and G. Hinton, “Deep Learning,” Nature, vol. 521, no. 7553, p. 436, 2015.

The HiPEAC project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement number 779656.

[3] L. Deng, J. Li, J.-T. Huang, K. Yao, D. Yu, F. Seide, M. Seltzer, G. Zweig, X. He, J. Williams et al., “Recent advances in deep learning for speech research at Microsoft,” in 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2013, pp. 8604–8608.

[4] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton et al., “Mastering the game of Go without human knowledge,” Nature, vol. 550, no. 7676, p. 354, 2017.

[5] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, “Deepdriving: Learning affordance for direct perception in autonomous driving,” in 2015 IEEE International Conference on Computer Vision. IEEE, 2015, pp. 2722–2730.

[6] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, “Efficient processing of deep neural networks: A tutorial and survey,” Proc. IEEE, vol. 105, no. 12, pp. 2295–2329, 2017.

[7] E. Wang, J. J. Davis, R. Zhao, H.-C. Ng, X. Niu, W. Luk, P. Y. Cheung, and G. A. Constantinides, “Deep Neural Network Approximation for Custom Hardware: Where We’ve Been, Where We’re Going,” arXiv preprint arXiv:1901.06955, 2019.

[8] NVIDIA. (2019) Automatic Mixed Precision for Deep Learning. [Online]. Available: https://developer.nvidia.com/automatic-mixed-precision

[9] Google. (2019) Using bfloat16 with TensorFlow models. [Online]. Available: https://cloud.google.com/tpu/docs/bfloat16

[10] Intel. (2018) BFLOAT16: hardware numerics definition. [Online]. Available: https://software.intel.com/sites/default/files/managed/40/8b/bf16-hardware-numerics-definition-white-paper.pdf

[11] C. De Sa, M. Leszczynski, J. Zhang, A. Marzoev, C. R. Aberger, K. Olukotun, and C. R´e, “High-accuracy low-precision training,” arXiv preprint arXiv:1803.03383, 2018.

[12] J.-M. Muller, N. Brunie, F. de Dinechin, C.-P. Jeannerod, M. Joldes, V. Lef`evre, G. Melquiond, N. Revol, and S. Torres, Handbook of Floating-Point Arithmetic, 2nd edition. Birkh auser Boston, 2018.

[13] C. Ding, S. Liao, Y. Wang, Z. Li, N. Liu, Y. Zhuo, C. Wang, X. Qian, Y. Bai, G. Yuan, X. Ma, Y. Zhang, J. Tang, Q. Qiu, X. Lin, and B. Yuan, “CirCNN: Accelerating and Compressing Deep Neural Networks Using Block-circulant Weight Matrices,” in Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture. ACM, 2017, pp. 395–408.

[14] K. Osawa, Y. Tsuji, Y. Ueno, A. Naruse, R. Yokota, and S. Matsuoka, “Large-scale Distributed Second-order Optimization Using Kronecker-factored Approximate Curvature for Deep Convolutional Neural Networks,” 2018.

[15] B. Barrois and O. Sentieys, “Customizing fixed-point and floating-point arithmetic A case study in K-means clustering,” in 2017 IEEE International Workshop on Signal Processing Systems (SiPS), Oct 2017, pp. 1–6.

The HiPEAC project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement number 779656.