PhD Subject Proposal
Precision Tuning for Deep Neural Networks Training

Supervisors: Silviu-Ioan Filip[1], Anastasia Volkova[2] and Olivier Sentieys[1]
[1]Univ Rennes, Inria; [2]Univ Nantes, LS2N
Campus de Beaulieu, 263 avenue du Général Leclerc, Rennes, France
silviu.filip@inria.fr, anastasia.volkova@univ-nantes.fr, olivier.sentieys@inria.fr

1 Context of the PhD Thesis

Deep Learning is one of the most intensively and widely used predictive models in the field of Machine Learning. Convolutional Neural Networks (CNNs) [1] have shown to achieve state-of-the-art accuracy in computer vision [2] 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 portofolio 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). Recent work [6] analyzing the carbon footprint of current natural-language processing models shows an alarming trend: training one huge Transformer model [7] for machine translation emits the same amount of CO_2 as five cars in their lifetimes (fuel included). 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 labelled 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 [8, 9]. 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 [10–12].

2 Objectives of the PhD Thesis

With this thesis, we want to conduct a thorough analysis of reduced numerical precision training of DL systems. We plan to do this at two levels: **arithmetic** (use appropriate numerical formats and bit widths for all the computations used during training) and **algorithmic** (by attempting to improve the practical convergence properties of the optimization procedures used to train neural networks).

A first objective 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.

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. The student will also have the task of validating the developed techniques trough a prototype of an accelerator for CNN training in the context of a collaboration with other researchers in the team.

2.1 Precision tuning for neural network optimization

Current research on neural network training acceleration focuses on low precision variants of SGD-based algorithms [13–17]. We plan to complement and improve on this existing work by investigating the use of adaptive numerical precision levels during the course of the training iterations[1].

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 [18]. Based on this, it might also prove worthwhile to explore the use of mixed-precision training for second-order DL methods.

Despite the success of backpropagation-powered training methods like SGD, there have been a number of concerns regarding them over the years. An important one is the vanishing gradient problem that results from the recursive application of the chain rule through consecutive layers of a deep network. Another problem is the fact that backpropagation does not allow simultaneous weight updates across layers, somewhat limiting parallel execution (although minibatch versions

[1] Increasing the precision as the algorithm converges is a common way to increase result accuracy.

of SGD offer opportunities for parallelization). Such problems motivate research into alternative training methods. Recent work focusing on alternating minimization methods is both welcome and promising [19–24]. We plan to also focus on numerical precision tuned versions of such algorithms.

2.2 CNN training acceleration through hierarchical modeling

Optimization of the training computations themselves by employing lower numerical precisions is not the only way to increase performance and improve energy efficiency. Due to the iterative nature of the algorithms used for learning, intelligent weight (parameter) initialization is also crucial for ensuring practical convergence. In particular, adequate starting values for the network weights limits the effects of exploding and/or vanishing gradients during backpropagation, which can also be very problematic when lowering the computational precision. Seminal papers dealing with effective weight initialization strategies include [25, 26] for symmetric activation functions and for the now more common rectifier linear unit (ReLU)-based activations.

In the case of CNNs, there is a current trend of employing network architectures that are scaled up [27–30] from some highly optimized base model in order to bypass earlier manual, time consuming network tuning labor. We think that such principled approaches for neural network architecture design can be enhanced with more principled weight initialization strategies as well (going beyond the aforementioned strategies of [25,26]). The basic idea we will explore is to train a hierarchy of smaller networks with similar topology much faster. This would hopefully allow us to extrapolate weight values for the much larger network that already offer a good testing accuracy, which can then be improved upon in a much smaller number of training iterations.

3 Host Team: Cairn@Inria

The 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 [31] and we consider them to also be applicable in complex DL systems as well. The team is working on integrating `ctfloat`[2], the custom low precision floating-point library for high-level synthesis we have developed, to the `N2D2`[3] 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.

References

[1] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep Learning. *Nature*, 521(7553):436, 2015.

[2] https://gitlab.inria.fr/sentieys/ctfloat/

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

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25 (NIPS 2012)*, pages 1097–1105, 2012.

[3] Li Deng, Jinyu Li, Jui-Ting Huang, Kaisheng Yao, Dong Yu, Frank Seide, Michael Seltzer, Geoff Zweig, Xiaodong He, Jason Williams, et al. Recent advances in deep learning for speech research at Microsoft. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8604–8608. IEEE, 2013.

[4] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of Go without human knowledge. *Nature*, 550(7676):354, 2017.

[5] Chenyi Chen, Ari Seff, Alain Kornhauser, and Jianxiong Xiao. Deepdriving: Learning affordance for direct perception in autonomous driving. In *2015 IEEE International Conference on Computer Vision*, pages 2722–2730. IEEE, 2015.

[6] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and Policy Considerations for Deep Learning in NLP. *arXiv preprint arXiv:1906.02243*, 2019.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[8] Vivienne Sze, Yu-Hsin Chen, Tien-Ju Yang, and Joel S Emer. Efficient processing of deep neural networks: A tutorial and survey. *Proc. IEEE*, 105(12):2295–2329, 2017.

[9] Erwei Wang, James J Davis, Ruizhe Zhao, Ho-Cheung Ng, Xinyu Niu, Wayne Luk, Peter YK Cheung, and George A Constantinides. Deep Neural Network Approximation for Custom Hardware: Where We've Been, Where We're Going. *arXiv preprint arXiv:1901.06955*, 2019.

[10] NVIDIA. Automatic Mixed Precision for Deep Learning, 2019.

[11] Google. Using bfloat16 with TensorFlow models, 2019.

[12] Intel. BFLOAT16: hardware numerics definition, 2018.

[13] Shuchang Zhou, Yuxin Wu, Zekun Ni, Xinyu Zhou, He Wen, and Yuheng Zou. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv preprint arXiv:1606.06160*, 2016.

[14] Shuang Wu, Guoqi Li, Feng Chen, and Luping Shi. Training and inference with integers in deep neural networks. *arXiv preprint arXiv:1802.04680*, 2018.

[15] Ron Banner, Itay Hubara, Elad Hoffer, and Daniel Soudry. Scalable methods for 8-bit training of neural networks. In *Advances in Neural Information Processing Systems*, pages 5145–5153, 2018.

[16] Naigang Wang, Jungwook Choi, Daniel Brand, Chia-Yu Chen, and Kailash Gopalakrishnan. Training deep neural networks with 8-bit floating point numbers. In *Advances in neural information processing systems*, pages 7675–7684, 2018.

[17] Christopher De Sa, Megan Leszczynski, Jian Zhang, Alana Marzoev, Christopher R Aberger, Kunle Olukotun, and Christopher Ré. High-accuracy low-precision training. *arXiv preprint arXiv:1803.03383*, 2018.

[18] Kazuki Osawa, Yohei Tsuji, Yuichiro Ueno, Akira Naruse, Rio Yokota, and Satoshi Matsuoka. Large-scale Distributed Second-order Optimization Using Kronecker-factored Approximate Curvature for Deep Convolutional Neural Networks, 2018.

[19] Gavin Taylor, Ryan Burmeister, Zheng Xu, Bharat Singh, Ankit Patel, and Tom Goldstein. Training neural networks without gradients: A scalable ADMM approach. In *International Conference on Machine Learning*, pages 2722–2731, 2016.

[20] Ziming Zhang, Yuting Chen, and Venkatesh Saligrama. Efficient training of very deep neural networks for supervised hashing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1487–1495, 2016.

[21] Akhilesh Gotmare, Valentin Thomas, Johanni Brea, and Martin Jaggi. Decoupling Backpropagation using Constrained Optimization Methods. In *Proc. of ICML 2018 Workshop on Credit Assignment in Deep Learning and Deep Reinforcement Learning*, 2018.

[22] Tim Tsz-Kit Lau, Jinshan Zeng, Baoyuan Wu, and Yuan Yao. A proximal block coordinate descent algorithm for deep neural network training. *arXiv preprint arXiv:1803.09082*, 2018.

[23] Thomas Frerix, Thomas Möllenhoff, Michael Moeller, and Daniel Cremers. Proximal backpropagation. *arXiv preprint arXiv:1706.04638*, 2017.

[24] Anna Choromanska, Benjamin Cowen, Sadhana Kumaravel, Ronny Luss, Mattia Rigotti, Irina Rish, Paolo Diachille, Viatcheslav Gurev, Brian Kingsbury, Ravi Tejwani, et al. Beyond Backprop: Online Alternating Minimization with Auxiliary Variables. In *International Conference on Machine Learning*, pages 1193–1202, 2019.

[25] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the 13th International Conference On Artificial Intelligence and Statistics*, pages 249–256, 2010.

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1026–1034, 2015.

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[28] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[29] Andrew G Howard et al. MobileNets: Efficient Convolutional Neural Networks For Mobile Vision Applications. *arXiv preprint arXiv:1704.04861*, 2017.

[30] Mingxing Tan and Quoc V Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *arXiv preprint arXiv:1905.11946*, 2019.

[31] 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)*, pages 1–6, Oct 2017.