An internship proposal within Labex Cominlabs LeanAI project

Mixed Precision DNNs Training with Second Order Optimization Methods

at ENS Lyon or Inria Rennes
Advisors: Elisa Riccietti (elisa.riccietti@ens-lyon.fr), Silviu Filip (silviu.filip@inria.fr)

Despite their massive success in training state-of-the-art deep learning models, gradient-based deep neural networks (DNNs) training methods such as stochastic gradient descent (SGD) still possess many weaknesses. Among them we have a low convergence rate that is heavily dependent on the tuning of the learning parameters, the well-known problem of vanishing and exploding gradients, and the fact that the backpropagation algorithm does not allow simultaneous weight updates across layers, somewhat limiting parallel execution. There has thus been significant interest in the development of alternative methods to tackle these difficulties and accelerate DNNs training.

In the context of this internship we wish to focus on two orthogonal and complementary strategies.

A first direction to obtain faster, less parameter dependent and more robust optimization methods, especially in case of highly non-convex problems, is to focus on higher-order optimization methods, which rely on derivative information beyond just first-order. These methods have been much less exploited in large-scale machine learning applications than first-order ones, due to their per-iteration cost and memory requirement, which render them unusable when faced with millions of parameters and training examples. Nevertheless, it has been shown that it is possible to devise variants of higher-order optimization methods that are as scalable as first-order methods and that attain provably faster convergence guarantees. Key ingredients to design such methods are the use of stochastic, diagonal or low-rank approximations of the Hessian matrix and subsampling techniques [2, 3].

A second approach to accelerate training is to use low precision arithmetic. Traditionally, DNNs training has been mostly done using 32-bit floating-point [1] arithmetic, but recent efforts have shown that in many cases it is possible to use lower precision computations (even going down to sub 8-bit floating-point formats) for SGD-based methods and still converge to an acceptable result [9–11].

If nowadays several attempts at introducing reduced precision in first-order training methods have been made, their use in second-order strategies is far less studied. Recent research has focused on employing second order information for quantization or pruning of neural networks. Some examples are HAWQ, CW-HAWQ [4, 5] or HAP [6], mixed-precision methods that exploit second order information (*e.g.,* by using fast approximations of the trace/eigenvalues of the Hessian) for an automatic selection of the relative quantization precision of each layer of a neural network or for structured pruning. Only preliminary results are however available for the use of multi-precision computations in second-order methods [7].

The **goal of this internship is** therefore **to explore the feasibility of low/mixed-precision second order DNNs training approaches**. In this sense, the student will write customized precision second order DNNs training algorithms (starting from the method presented in [2]) and explore the convergence properties of such an implementation. The work will build upon a custom precision training simulation framework constructed atop PyTorch, called `mptorch`, developed by the internship coordinators.

The ideal candidate must be familiar with continuous optimization methods (first and second-order) and how stochastic versions of these algorithms are used in supervised deep learning applications. Experience with Python programming is also required. Knowledge of deep learning frameworks such as Pytorch is also desirable.

References

[1] J.-M. Muller, N. Brunie, F. de Dinechin, C.-P. Jeannerod, M. Joldes, V. Lefèvre, G. Melquiond, N. Revol, and S. Torres. *Handbook of Floating-Point Arithmetic*. Birkhäuser Boston, 2nd edition, 2018.

[2] Z. Yao, A. Gholami, S. Shen, K. Keutzer, and M. W. Mahoney. AdaHessian: An Adaptive Second Order Optimizer for Machine Learning. *arXiv preprint arXiv:2006.00719*, 2020.

[3] F. Roosta-Khorasani and M. Mahoney. Sub-sampled Newton Methods. *Math. Program.*, 174:293–326, 2019.

[4] Z. Dong, Z. Yao, A. Gholami, M. W. Mahoney, and K. Keutzer. Hawq: Hessian aware quantization of neural networks with mixed-precision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 293–302, 2019.

[5] X. Qian, V. Li, and C. Darren. Channel-wise Hessian Aware trace-Weighted Quantization of Neural Networks. *arXiv preprint arXiv:2008.08284*, 2020.

[6] S. Yu, Z. Yao, A. Gholami, Z. Dong, M. W. Mahoney, and K. Keutzer. Hessian-aware pruning and optimal neural implant. *arXiv preprint arXiv:2101.08940*, 2021.

[7] S. Gratton and P. Toint. A note on solving nonlinear optimization problems in variable precision. *Comput Optim Appl*, 76:917–933, 2020.

[8] A. S. Berahas and M. Takáč. A robust multi-batch L-BFGS method for machine learning. *Optimization Methods and Software*, 35(1):191–219, 2020.

[9] N. Wang, J. Choi, D. Brand, C.-Y. Chen, and K. Gopalakrishnan. Training deep neural networks with 8-bit floating point numbers. In *Advances in Neural Information Processing Systems*, pages 7675–7684, 2018.

[10] N. Mellempudi, S. Srinivasan, D. Das, and B. Kaul. Mixed precision training with 8-bit floating point. *arXiv preprint arXiv:1905.12334*, 2019.

[11] P. Zamirai, J. Zhang, C. R. Aberger, and C. De Sa. Revisiting BFloat16 Training. *arXiv preprint arXiv:2010.06192*, 2020.