An internship proposal within Labex Cominlabs LeanAI project

Mixed Precision SGD strategies

at ENS Lyon or Inria Rennes

Advisors: Elisa Riccietti (elisa.riccietti@ens-lyon.fr), Silviu Filip (silviu.filip@inria.fr)

Stochastic gradient descent (SGD) and its variants are the basis of the state-of-the-art techniques used for training deep learning models. However, for large deep networks, training using such gradient-based methods has a high computational and energy cost associated with it.

A recent research direction to speed up the training of deep neural networks (DNNs) and improve its energy footprint is the use of low precision arithmetic. Traditionally, DNN training has been mostly done using 32-bit floating-point [7] 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 [6, 9, 10]. Most of the existing methods in the literature just fix a low precision for the network and hope for the training to converge, which is not always the case. A more interesting research direction is to rather use variable precision in the computations, thus increasing the precision when needed, in order to guarantee convergence in all cases. The major challenge associated with these methods is to decide when and where to increase the precision.

The **aim of this internship will be to study variants of SGD based on an automatic choice of the precision for the computations, able to guarantee convergence in all cases**. To achieve this, we will explore the framework of trust region methods.

Trust region schemes have been successfully employed in classic and stochastic optimization [1–4] as an easy and automatic way of selecting a step-size for optimization methods, which ensures convergence of the method for any initial iterate. This mechanism can be employed as well to select the precision of the computations, with preliminary work done in a deterministic setting [5] for second order methods. The extension to a stochastic context and to first order methods, a setting that is more suitable for neural networks training, is still an open question.

In this internship we will implement and study a mixed-precision SGD method based on a trust-region scheme to automatically update the precision. The work will build upon a custom precision training simulation framework constructed atop PyTorch, called `mptorch` [8], developed by the internship coordinators.

The ideal candidate must be familiar with continuous optimization methods 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] Convergence rate analysis of a stochastic trust-region method via supermartingales. *INFORMS, Journal on Optimization*.

[2] S. Bellavia, N. Krejić, B. Morini, and S. Rebegoldi. A stochastic first-order trust-region method with inexact restoration for finite-sum minimization. *Computational Optimization and Applications*, pages 1–32, 2022.

[3] R. Chen, M. Menickelly, and K. Scheinberg. Stochastic optimization using a trust-region method and random models. *Mathematical Programming*, 169(2):447–487, 2018.

[4] F. E. Curtis and K. Scheinberg. Adaptive stochastic optimization: A framework for analyzing stochastic optimization algorithms. *IEEE Signal Processing Magazine*, 37(5):32–42, 2020.

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

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

[7] 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.

[8] M. Tatsumi, Y. Xie, C. White, S.-I. Filip, O. Sentieys, and G. Lemieux. MPTorch and MPArchimedes: Open Source Frameworks to Explore Custom Mixed-Precision Operations for DNN Training on Edge Devices. In *ROAD4NN 2021-2nd ROAD4NN Workshop: Research Open Automatic Design for Neural Networks*, 2021.

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