Open Thesis / Project

Enabling N:M Sparsity Without Penalty

(Embedded Information Processing Team)

Motivation

Due to large neural network size and high redundancy, there is a growing interest in various techniques to reduce the number of weights and accelerate training and inference. An active area of research in this field is sparsity - encouraging zero values in parameters that can then be discarded from storage or computations. The recently introduced NVIDIA Ampere accelerator architecture supports 2:4 sparsity pattern, *i.e.*, halves a model's parameter count, requiring that every group of consecutive four values contains at least two zeros. This leads to twice the math throughput of dense matrix units. In general, we consider N:M sparsity. In this work, your will explore how to efficiently build, zip and update N:M-sparse networks. As a starting point, we encourage you to read the following papers: (1) https://drive.google.com/file/d/1IL-XxR6C1IA1evN-QJ8lw1n6Y2XSQbH_/view, (2) https://arxiv.org/abs/2104.08378, (3) https://arxiv.org/abs/1805.09791. You will be provided an algorithm helpful in this setting. Your ideas are welcome. **Interested? Contact us for more details!**

Target Group

Students in ICE and Computer Science.

Thesis Type

Master Project / Master Thesis.

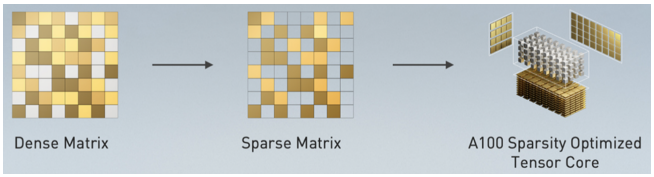

Image source: https://bit.ly/3zND1gw

NVIDIA Ampere architecture providing computational and storage speed-up for weight matrices with 2:4 sparsity.

Goals and Tasks

In this project, after sparsifying the network with a pruning method of your choice, one has to re-arrange the columns in the weight matrix to follow the desired N:M pattern. We already have ideas how to do this and we will ask you to test method efficiency. The same approach can also be effective in the context of multi-task zipping and collaborative training. You will explore these possibilities and compare to the methods described in the literature. The project includes the following tasks:

- Literature review on N:M sparsity, permutation invariance, network compression, multi-task zipping and distributed training;
- Implement N:M sparsity and compare to the vanilla algorithm published in the literature;
- Evaluate the same idea in the context of multi-task zipping and distributed training;
- Summarize the results in a written report, and prepare an oral presentation.

Requirements / Skills:

- A good knowledge of neural networks and interest in exploring network sparsity;
- Programming skills in Python;
- Prior experience in deep learning frameworks is desirable (preferably PyTorch).

Used Tools / Equipment:

- A laptop (GPU infrastructure will be provided)
- Your talent (very important!).

Contact Person

- Rahim Entezari (entezari@tugraz.at)
- Dr. Olga Saukh (saukh@tugraz.at)

Institute for Technical Informatics
Networked Embedded Systems Group