Data Compression for Machine-Learning

Description

Machine-Learning (ML) applications are an emerging type of application with huge demands on both compute power and memory resources. In this project we will focus on optimizations for the memory resources. It is known that ML applications put a considerable pressure on the memory bandwidth when gathering inputs for the Neural Network implementation as well as memory storage for the intermediate results. Reducing the pressure on the input bandwidth would result in lower latency for the execution while reducing the internal storage for intermediate results would result in being able to store more intermediate results and thus achieve better optimizations or alternatively reduce the physical resources and achieve saving in power consumption.

One proven way to reduce memory bandwidth and storage pressure is to apply data compression. Several efficient data compression techniques have been developed in the past with impressive results.

Depending on the interests of the student(s) there can be more emphasis on certain steps, for example a student more interested on hardware should focus more on the HW implementation while a student more interested on software should focus more on the analysis and algorithms for this project. A team with two students, one focused on the software aspects and one focused on the hardware aspects would also be accepted.

Goals

In this project we would like to study how data compression could be used to improve the performance of ML execution. We are interested in learning more on:

- Which memory compression techniques are a better match to the ML data?
- Which compression can be achieved for the input bandwidth? And which overhead is to be expected?
- Which compression can be achieved for the internal intermediate result storage space? And which overhead is to be expected?
- Are there different requirements (and achieved results) for compressing training data versus inference data?
- How does compression change for different levels of quantization? Is it as effective for 32b FP as for 8b int quantized models?
- How does compression change for different degrees of pruning of a model? Compression in sparce models is as effective as in dense models?
- Is there a possibility of using lossy compression instead of lossless compression techniques? (impact on accuracy)
- What is the hardware overhead for performing the data compression?

Not all questions will be answered by a single project. Multiple projects can be defined from this same topic.

Pre-requisites

For the success in this project the student needs to have basic knowledges of computer architecture. Basic knowledge of Machine-Learning is an added benefit but not a requirement.

Methodology

The execution of this project will go through the steps below (as mentioned below, depending on the focus of the project, there will be more emphasis on the software or the hardware aspects):

- Selection of a ML application – selection of ML model and platform (training of a model is not required, pre-trained models should be used when necessary)
- Clear identification of the problem and formulation of the research question (the research question should come from the questions stated above in the description)
- Preliminary experiments to justify the problem
- Analysis of the state-of-the-art in ML compression and existing data compression algorithms.
- Introduction of compression into the ML flow
- Evaluation of different compression techniques
- Selection of a compression technique and proposal of a simple HW module to perform the compression
- Evaluation of the HW compression module for the target application

Selected Relevant References

- Song Han, Huizi Mao, William J. Dally, "Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding", 2016, arXiv:1510.00149
- Angelos Arelakis, Fredrik Dahlgren, and Per Stenstrom. 2015. HyComp: a hybrid cache compression method for selection of data-type-specific compression methods. In Proceedings of the 48th International Symposium on Microarchitecture (MICRO-48). Association for Computing Machinery, New York, NY, USA, 38–49. DOI:https://doi.org/10.1145/2830772.2830823
- Yousun Ko, Alex Chadwick, Daniel Bates, and Robert Mullins. 2021. Lane Compression: A Lightweight Lossless Compression Method for Machine Learning on Embedded Systems. ACM Trans. Embed. Comput. Syst. 20, 2, Article 16 (March 2021), 26 pages. https://doi.org/10.1145/3431815

Target group

DV, D, E and IT

Contact:

Pedro Petersen Moura Trancoso
Computer Science and Engineering
ppedro@chalmers.se