EPFL http://esl.epfl.ch Embedded Systems Laboratory (ESL)
Director: Prof. David Atienza Alonso

Weight scaling for overlapping weights encoded in a transpose pattern

Contact Person: Mr. William Simon (william.simon@epfl.ch),
Dr. Alexandre Levisse (alexandre.levisse@epfl.ch),
Prof. David Atienza (david.atienza@epfl.ch)

Project Description

Over the last decade, neural networks have been applied to a wide range of applications with great success. However, the weights they store result in a large memory overhead at all levels of the memory hierarchy [1]. Much work has been done to mitigate the drawbacks of this characteristic of neural networks.

The goal of this project is to continue to explore the utilization of a new transposable access RRAM memory to increase the storage capacity allocated to weights, reducing the necessity for memory movement.

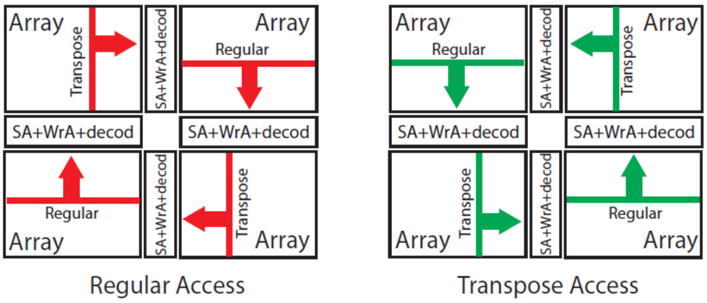

Figure 1: Transposable access memory

Figure 1 presents a high-level view of the functionality of the transpose access memory array. At each memory access, each subarray can be accessed in either a regular or transposed manner, enabling new data access patterns that can accelerate application runtime.

In order to take advantage of such an architecture for the acceleration of neural networks, we encode neural network weights such that they overlap in a regular/transpose manner, effectively doubling the capacity of the memory array at the cost of a loss of weight accuracy (not necessarily neural network accuracy however). Weights are stored such that the MSBs of weights stored in regular fashion overlap with the LSBs of weights stored in transpose fashion. Such a configuration reduces the error induced by bit sharing. Individual bit values are then quantized to minimize the error of the final weights with respect to their original values. This adjustment of weight values, combined with network retraining via incremental retraining[2], enables more compact weight storage at little to no loss of neural network accuracy.

One open question discovered during the research process is how weight scaling during quantization from floating- to fixed-point impacts network accuracy. When a layer of floating-point weights is quantized to fixed point, the values are scaled between the min and max fixed-point values (ex 0 to 255 for 8-bit fixed point.) Performing a 1-to-1 scaling in which the min/max floating-point values are set to the min/max fixed-point values does not result in the highest possible network accuracy, as most weight values are congregated around 0. It is instead more effective to set a lower weight value as the max fixed-point value, and clamp all weights higher than that value to the max value. This results in some weights losing accuracy, but induces more variance in the weights, thus improving accuracy. This has been demonstrated empirically, but an algorithmic solution is desired.

Tasks of the Student:

1. Literature exploration of current scaling methods.
2. Develop a strategy for selecting an optimal scaling value for weight quantization.
3. Implement the algorithm for weight scaling in the current PyTorch framework utilized by this ongoing project.
4. If previous objectives are fulfilled, this work will be a part of a submission in a peer-reviewed conference or journal.

Requirements:

- Understanding of neural network theory, both training and inference.
- Previous work with Matlab, Python or C++.

Appreciated Skills:

- Previous work with PyTorch or equivalent neural network framework.
- Motivation to learn new skills.
- Autonomous work ability.

References:

[1] S. Bianco, R. Cadene et al., “Benchmark analysis of representative deep neural network architectures,” IEEE Access, vol. 6, 2018.

[2] Incremental Network Quantization: Towards Lossless CNNs with Low-Precision Weights