Extended Bit-Plane Compression: Alleviating the Costs of Data Transfer for Edge AI

Georg Rutishauser, Ph.D. Student, Integrated Systems Lab, ETH Zurich

Various factors, such as power, communication bandwidth, latency constraints, and privacy concerns, make it attractive to analyze the data captured by various sensors device right where it is collected – a trend commonly summarized under the term "Edge AI." Specialized hardware accelerators make the computations highly energy-efficient, leaving the memory accesses the dominant overall contributor to power consumption. Reducing these accesses to central memory thus becomes a central objective in the quest for energy-efficient Edge AI devices.

To reduce the volume of data that needs to be stored and transferred, we propose *Extended Bit-Plane Compression (EBPC),* a lossless, variable-length compression scheme targeted at DNN feature maps. It achieves state-of-the-art compression ratios on many well-known deep neural networks while remaining suitable for hardware implementation. The algorithm is a fusion of two established methods, taking advantage of the *smoothness* and *sparsity* inherent to CNN feature maps: 1) bit-plane compression (BPC), a compression scheme originating from texture compression which exploits data smoothness, and 2) zero run-length encoding (ZRLE), exploiting data sparsity. It encodes data in two compressed streams: a zero-nonzero bitstream compressed using ZRLE, indicating for each input word if it is zero, and a BPC stream, which is the result of applying BPC to the nonzero values.

Our experiments demonstrate average compression ratios between 2.2x (MobileNetV2) and 5.1x (AlexNet) using activations quantized to 8 bits. The encoder and decoder implementations each synthesize to around 3'000 gates for 8-bit data. Post-layout power simulations in GF 22nm at 556 MHz (523 Mword/s) show an energy cost of less than 0.4 pJ/word. This enables the compression method to not only reduce off-chip transfer energy (from ~160 pJ/word to ~42 pJ/word) and to reduce the memory size, but to save upwards of 35% energy when storing feature maps in on-chip SRAM relative to uncompressed read/write operations.