IEEE International Symposium on Workload Characterization – 2022

Title: Sparse Weight Compression and Decompression for Intel AMX/TMUL to Improve Deep Learning Performance

Authors: Shamima Najnin Rajesh Poornachandran, Sreekanth V. Yalachigere Mona Minakshi, Anik Khan, Ofir Zafrir, Nilesh Jain, Md Faijul Amin, Guy Boudoukh, Tatyana Primak, Pallavi G, Intel Corporation

Abstract:
Recent trends in deep neural network (DNN) models designed with billions of parameters exhibits human level capabilities, but at the cost of significant computational and memory resources (e.g., memory, CPU, energy, bandwidth (BW)). This makes model deployment challenging and expensive in the production environment of datacenter as well as in wide variety of other applications such as virtual reality, augmented reality, smart wearable devices etc. Intel® Deep Learning Boost (Intel® DL Boost) roadmap has been making continuous improvement in the compute capabilities, but memory BW has not grown at the same rate as compute. Memory is becoming an expensive platform component and limiting compute, thereby we're at a tipping point to optimize memory to unleash max compute capabilities with TCO Savings (i.e. perf/W/$). Key cloud customer workloads being accelerated on Xeons (Broadwell (BDW) to Cascade Lake (CLX) to Ice Lake (ICX) to Sapphire Rapids (SPR)), involve Convolution Neural Network (CNN) models and larger Fully Connected (FCs) dominated models that benefit from wider SIMD width/Tiled structure acceleration offered by Xeons. However, Recommender systems, Natural Language Processing (NLP) use cases involving Multilayer-perceptron (MLP)/Long-short term memory (LSTM) shifts the bottleneck from compute to memory bound wherein these use cases are ~90% of AI inference workload mix in datacenters. From BDW to SPR, compute scaling (INT8 FLOPs) has increased 64x whereas memory scaling (theoretical max memory BW) is only 4x.

Recently, Sparsity is gaining popularity to reduce the memory footprint of ever growing DNNs. Compressing the sparse models provides an opportunity to address compute/memory scaling challenges as well as shorten training/inference time to maximize the performance on resource and energy-constrained target hardware platforms. Pruning (introduction of sparsity) and quantization algorithms can help to compress neural networks by an order of magnitude. Recent DNN pruning research has shown that 90% of the weights can be removed in unconstrainted scenario with little loss in accuracy which resulted in unstructured sparse weight matrices. Though unstructured sparsity provides most flexibility, it is hard to map efficiently on modern CPUs and GPUs with limited support in hardware and software packages. This limitation leads to development of structured sparsity, but unfortunately inclined to higher accuracy loss than unstructured sparsity. Software approaches for unstructured sparsity can help to a certain extent but not completely and aren't efficient or pragmatic for volume product intercept with TCO

advantages. A notable aspect of sparsity is that it's not only a tool to optimize computation resources, but also has been found to be efficient representation in primate brain to process visual information. This solution is first for Xeon Sparsity in SW for SPR and potential HW intercept in future.

During inference runs, memory bandwidth (BW) (both DRAM and Cache) becomes bottleneck for inner product between large matrices (e.g., BERT_Large) weights cannot be cached due to large number of parameters; cores are starved while waiting for the data. This increases cache thrashing and misses per instructions (MPI), requiring higher memory BW resulting in lower performance.

We have proposed solution as sparse weight compression/Decompression in Inner-Product kernel of oneDNN to address this challenge. In this solution, 90% of the weights can be pruned with little loss in accuracy which resulted in unstructured sparse weight matrices and weights are reordered and zero-compressed to save memory footprint. During execution, compressed weights are read from DRAM to LLC (lower memory BW) and decompressed using AVX512 vpexpandb. Then, the decompressed weights in L1 cache are loaded to AMX tiles for TMUL Operations (Lower L3-L2/L1 BW).
To benchmark our proposed solution, we have considered BERT-Large real-time performance with batch size=1 with multi-instance approach with NUMA and Memory binding on 4th Gen Intel® Xeon ® Scalable Processor (Sapphire Rapids). We conducted performance analysis by varying sparsity level and sequence length and have achieved the performance gain using the proposed solution for 70%, 80% and 90% int8 sparse model with varying sequence length and have observed significant performance gain with increase of sparsity due to higher compression capability leading to improved TMUL utilization & memory BW savings. Performance gain increases with smaller sequence length as compute density lowers with more memory access, i.e., memory boundness where compression helps to efficiently utilize limited memory BW. This has also helped in BW savings with varying sequence length for the same 90% sparse model. It can be demonstrated that using proposed solution read memory BW can be saved up to ~1.8x whereas total memory BW (read + write) can be saved up to ~1.7x which results up to 1.7x inference throughput speedup. It is also found that proposed solution helps to reduce memory boundness by retiring more instructions, thereby reducing cache miss rates alongside DRAM memory BW improvement. Larger networks appear to provide more opportunity for pruning so the compression trend is likely to continue as architectures get larger. Our detailed analysis above enables a targeted hardware-software co-design for next-generation deep learning architectures that exploit the potentially huge speedups. For instance, AVX512 is used to decompress one cache line at a time for decompression in this solution.