Paper Reading Report

曹金坤 515260910022

The paper *"Neural Cache: Bit-Serial In-Cache Acceleration of Deep Neural Networks"* focuses on the efficiency optimization on memory-level targeting for the operations in deep neural networks. To be precise, this paper makes following contributions:

1. The paper presents a design to compute SRAM array, for which addition and multiplications are supported.
2. The paper re-purposes design to do in-situ computation in caches to save the on-chip data movement overheads.
3. The paper presents a so-called ``Neural Cache" architecture to re-purpose the last-level cache (LLC) to accelerate Deep Neural Networks inferences, for which specific parallel computation is exploited. And a novel data layout scheme is designed to support this accelerations.

Besides, the proposed design can support multiple DNN layers, including fully connected layer, pooling layer and convolutional layers. Related operations are expected to be performed mostly in cache to leverage better efficiency.
To save the report page, I choose some key contributions and explain them.

Neural Cache Arithmetic: bit-serial arithmetic and transposed data layout are combined to do faster cache arithmetic. In such scheme, arithmetic operations are performed on bit-level-parallelism for the different data lines. It is supported in addition, multiplication and reduction, which are common in DNN inferences. The other main novelty for neural cache arithmetic is the transposed data layout, which is based on transpose gateway units. To be precise, designed hardware transpose memory units (TMUs) take data in the bit-parallel or regular layout and convert it to the transpose layout before storing into the SRAM arrays or vice-versa during reading from SRAM. This design connect the bit-parallel arithmetic and traditional SRAM operation.

Neural Cache Architecture: The neural cache architecture transforms SRAM arrays in LLC to compute functional units, which helps accelerate DNN operations. It exploits channel level parallelism in a single convolution. The design of this architecture conforms to the convolution conventions, such as the transposing of data channel and batch size. On the other hand, for convolution operation, feature maps would be sliced into pieces and then convoluted as usual, which makes the parallel convolutions possible. At last, this architecture introduces matching data movement and data management scheme, which are both adapted to the transpose data layout and parallel data computation proposals.

In conclusion, this paper proposes novel management and computation scheme for data in memory/cache. To better leverage computation potential and decrease data movement overheads in DNN operations, it specifically designed transpose data layout and parallel computation supporting most common operations in DNN inference. The transpose data layout conforms to the data layout in convolution and parallel computation comes from the data slicing and special data line management. Experiments on popular benchmark settings proved the efficiency of proposed design.