Efficient Deployment of Very Wide and Very Deep Hypersparse FFNs on FPGA

Paramdeep Singh (psingh7@scu.edu)
Advisor: Dr. David C. Anastasiu (danastasiu@scu.edu)

Background

- Feed Forward Networks (FFNs) are an integral part of modern cognitive networks such as CNN's and Transformer based LLMs.
- The advent of the Transformer architecture has renewed the focus on efficient processing of FFN's.
- Fully connected layers in these networks can account for up to two-thirds of floating-point operations (FLOPS) during the inference step.
- FFNs can be compressed using aggressive pruning and quantization.

Problem Description

In pruned networks represented using CSR data structure, the memory requirement of indices arrays alone can far outstrip memory requirement of model parameters.

PROPORTION OF MEMORY OCCUPIED BY INDICES ARRAYS

Quantization (bits)	Network Width 1024	2048	4096	8192
16	38.46%	40.74%	42.86%	44.83%
8	55.56%	57.89%	60.00%	61.90%
6	62.50%	64.71%	66.67%	68.42%
4	71.43%	73.33%	75.00%	76.47%

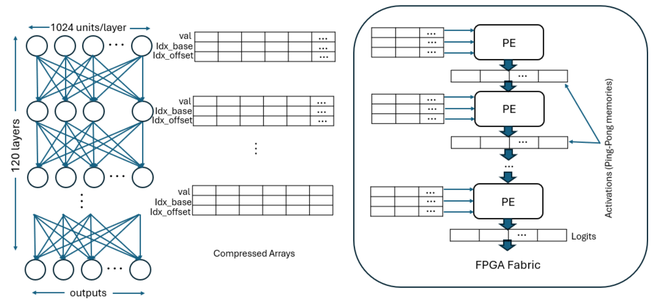
Objectives

- Minimize off-chip memory accesses by reducing total memory requirement (parameters + CSR structures) of the model.
- Optimize Usage of scarce BRAM and URAM as much as possible.
- Parallelize execution of Processing Elements (PEs)

Overall Approach

- Radix-Net pre-pruning.
- Compress indices arrays in software using the encoding algorithm.
- Decompress the indices array in hardware in the Processing Element (PE).
- Implement layers as a cascade of PEs with Dataflow pipelining.

System Architecture



Encoding Algorithm

array of absolute indices
(bit-width of each index = $\log_2$(Network Width)

00011 11011	00011 11011	00100 00101	00110 10000	00110 10111	00111 01111	00111 11101	01000 10001
01000 11000	01001 01110	01001 10000	01010 11010	01100 11000	01101 10010	01111 00110	01111 01100
01111 10100	10000 01111	10001 00000	10001 10100	10011 11010	10100 01111	10101 11000	10110 10100
10111 00011	11000 00000	11001 00010	11010 10101	11011 01100	11100 10111	11100 11011	11101 11000

idx_offset array

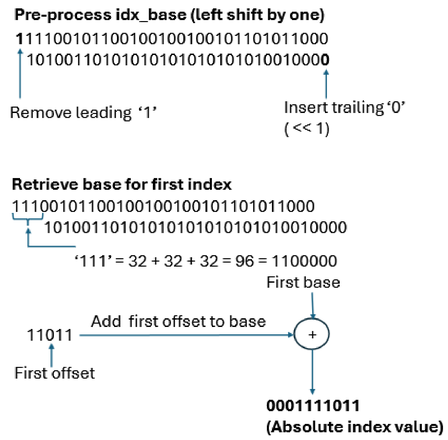
11011	11011	00101	10000	10111	01111	11101	10001
11000	01110	10000	11010	11000	10010	00110	01100
10100	01111	00000	10100	11010	01111	11000	10100
00011	00000	00010	10101	01100	10111	11011	11000

bitmap corresponding to neuron
(single entry in idx_base array)

1111001011001001001001011010110001010011010101010101010101001000

Initialization bit

Decoding Algorithm



Results

- Method tested on the Xilinx VC 709 board.
- Results comprise two parts:
  - Efficiency results
  - Effectiveness Results

Efficiency results

- Inference Performance

Method	Performance Latency	Throughput
Baseline (Matlab)	124.07	N/A*
Baseline (Huang et. al.)	251.31	N/A*
Ours	0.3	0.0024

*N/A = Throughput not reported

- Resource Usage

Method	Resource Usage LUTs	FFs	BRAMs	DSPs
Baseline (Matlab)	N/A*	N/A*	N/A*	N/A*
Baseline (Huang et. al.)	48.43%	26.86%	55.44%	4.17%
Ours	4.45%	2.30%	93.88%	0%

*N/A = Not reported

Effectiveness results

- Accuracy (%)

# Layers	Fully Connected (32b)	Pruned (32b)	Pruned (4b)
3	96.81	97.45	95.88
6	95.87	98.08	97.57
10	95.53	97.95	97.82
20	18.17	95.91	94.73
30	9.8	9.8	9.8
40	9.8	9.8	9.8
50	9.8	9.8	9.8
60	9.8	9.8	9.8
70	9.8	9.8	9.8
80	9.8	9.8	9.8
90	9.8	9.8	9.8
100	9.8	9.8	9.8
110	9.8	9.8	9.8
120	9.8	9.8	9.8

Future Work

- Construct Radix-Net topologies guided by unstructured pruning.
- Replace indices arrays completely with state machines .

Acknowledgements

Research supported by a Supermicro GPU SuperServer SYS-420GP-TNAR+ node contributed by Supermicro and NVIDIA, integrated into the Santa Clara University HPC.