TRIP: An Ultra-Low Latency, TeraOps/s Reconfigurable Inference Processor for Multi-Layer Perceptrons

Ahmed Sanaullah[1], Chen Yang[1], Yuri Alexeev[2], Kazutomo Yoshii[3], Martin Herbordt[1]

[1]CAAD Lab, Boston University (USA); [2]Leadership Computing Facility, Argonne National Laboratory (USA); [3]Mathematics and Computer Science Division, Argonne National Laboratory (USA)

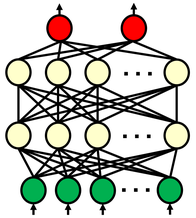
Abstract

Multi-Layer Perceptron (**MLP**) is one of the most commonly deployed Deep Neural Network, representing 61% of the workload in Google data-centers [1]. MLPs have low arithmetic intensity which results in memory bottlenecks. To the best of our knowledge, the Google Tensor Processing Unit (TPU) [1] is currently the state-of-the-art implementation of MLP inference. TPU addresses the memory bound by processing multiple test vectors simultaneously to increase operations per weight byte loaded from DRAM. However, inference typically has hard response time deadlines and prefers latency over throughput [2]. As a result, waiting for sufficient input vectors to get good performance is not feasible. In this work, we designed a **TeraOps/s Reconfigurable Inference Processor (TRIP)** for MLPs on FPGAs that alleviates the memory bound by storing all weights on-chip and ensures performance is invariant of input batch size. For large databases that cannot directly fit on chip, Deep Compression [3] relaxes memory footprint requirements with no effect on accuracy. TRIP can be deployed as a standalone device directly connected to data acquisition devices, as a co-processor where input vectors are supplied through OpenCL wrappers from the host machine, as well as in a cluster configuration where on-chip transceivers can communicate between FPGAs. By comparison, TPU can only be used in a co-processor configuration. Our design achieved 3.0 TeraOps/s and 1.49 TeraOps/s on Altera Arria 10 for Stand-Alone/Cluster and Co-Processor configurations respectively, making it the fastest real-time inference processor for MLPs.

Multi Layer Perceptron

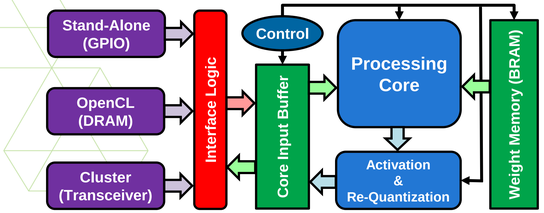
- Fully connected layers of neurons
- Layer inputs are non-linear functions of the sum of scaled neuron outputs of the previous layer
- Typically asymmetric logical configuration
- Performance has higher dependency on intra-layer operations
- Memory bound: no weight reuse for a test vector
- Inference can be performed in fixed point without loss of accuracy

Our design uses 8 bit quantization and 32 bit activations



Architecture Overview

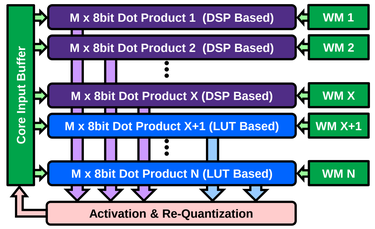
- Designed for deployment in Stand-Alone, Coprocessor and Cluster configurations
- Interface logic is deployment specific: GPIO for standalone, OpenCL wrappers for co-processor and transceivers + router for cluster
- Core input buffer contains a FIFO for incoming test vectors and a buffer to store intermediate results
- Processing Core contains multiple quantized MAC units for performing scalar products
- Weights are supplied from on-chip BRAM
- Activation and Re-Quantization module applies ReLU activation and converts 32 bit results into 8 bit
- The control state machine parameters are initialized at configuration time. No external instructions are required



Architecture Details

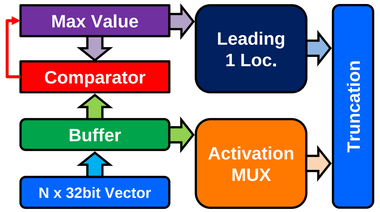
Processing Core

- Implemented with up to 8192 – 8bit integer multipliers in a ***M*** x ***N*** 2D array
- Both DSP and ALM multipliers are employed
- Each slice of ***M*** multipliers has an adder tree to evaluate scalar product and is supplied weights by an independent Weight Memory module
- Application specific values of ***M*** and ***N*** to maximize useful OPs



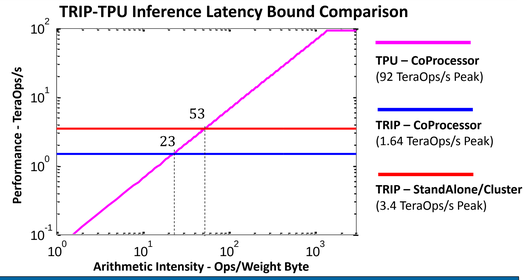
Activation Pipeline

- Max value of result vector used to truncated 32 bit result to 8 bits
- Max value search performed as a series of local maxima searches
- Short pipelines with logarithmic complexity to reduce inter-layer latency
- ReLU activation done using MUXs with sign bit as select
- Internal buffer used to accumulate and store partial sums



TRIP vs Tensor Processing Unit (TPU)

- TPU MLP implementation is memory bound due to slow off-chip memory access
- By processing multiple test vectors simultaneously, weight re-use in TPU improves performance and the cost of inference latency
- We compare TPU and TRIP to determine the inference latency bound needed by TPU to outperform our design
- The first generation of TPU has 64K MACs, 30GB/s off-chip bandwidth, and 700MHz operating frequency
- TRIP is deployed with 8192 multipliers for stand-alone/cluster, and 4096 multipliers for co-processor designs. Operating frequency is 200MHz
- Input data fetch latency is assumed to be negligible
- From the figure, we estimate that TRIP outperforms TPU for input batch sizes of less than 53 test vectors



ECP-CANDLE Benchmark

- We have used the ECP-Candle-P3B1 Tumor Laterality (TL) and Clinical Grade of Tumors (CGT) benchmarks to test our system
- The MLP model is composed of four layers with 400 input neurons. Number of output neurons are 2 for TL and 4 for CGT
- Hidden layer dimensions are (400,1200), (1200,1200), (1200,1200) and (1200,2 or 4) respectively. Weights and biases are trained offline and quantized to 8 bits
- Quantization with truncation has an error of 0.98% for TL and 1.8% for CGT with respect to floating point implementations

Hardware Specification

Type	Name	Description
GPU	NVIDIA TESLA K80	49992 CUDA cores - 480 GB/s Global Memory Bandwidth – cuBLAS library – CUDA 8.0
FPGA	ALTERA ARRIA 10X1150	427,200 ALMs - 1518 DSP Blocks (2 18x18 Integer Multipliers per block) – 54,260 Kb BRAM

Results

FPGA Parameters

- Table 1 lists the FPGA utilization summary for different deployment configurations
- All designs implement 10 DSP based MAC units
- 8192 multipliers for stand-alone/cluster, and 4096 multipliers for OpenCL co-processor designs
- Resource utilization for Co-Processor includes OpenCL wrapper logic
- Cluster resource usage includes transceiver and router logic

Table 1: TRIP Deployment based FPGA Parameters

Configuration	ALM	DSP (block)	BRAM (MB)	Freq. (MHz)
Stand-Alone	313,294 (73%)	1280 (84%)	4.0 (60%)	207
Co-Processor	250,710 (59%)	1280 (84%)	4.3 (65%)	201
Cluster	314,794 (74%)	1280 (84%)	4.0 (60%)	206

Performance

- ECP-Candle input batch size = 1
- Performance values in Table 2 are the best case from TL and CGT results
- GPU: cuBLAS MVM and custom activation kernel
- Stratix 10: 4x more ALM multipliers, 4 more DSP scalar product slices and 2x operating frequency
- TPU: performance estimated based on memory bandwidth constraints
- Input vector access latency is ignored for our estimations since it can be masked by computations
- TRIP is **orders of magnitude faster** than TPU for single input vectors
- TRIP has better resource utilization (**10% more useful ops**) due to variable Processing Core dimensions

Table 2: ECP-Candle Performance Comparison for Single Input Vector

Inference Architecture	M,N	Useful Op (%)	Performance (TeraOps/s)	Speedup
NVIDIA K80	-	-	0.02	1x
TPU	256,256	79	0.05	2.5x
TRIP Arria 10 CoProc	256,16	91	1.5	75x
TRIP Arria 10 Cluster	256,32	89	3.0	150x
TRIP Stratix 10 CoProc	256,86	88	15.5	775x
TRIP Stratix 10 Cluster	256,102	86	18.0	900x

Power

Table 3: TRIP-TPU Power Comparison

Architecture	Static	Dynamic	Total
TPU	-	-	38-43W
TRIP CoProc	30W	2W	32W

Impact

- To the best of our knowledge, TRIP is the only TeraOps/s MLP inference engine for small input batch sizes
- TRIP's deployment versatility and low power consumption makes it an ideal candidate for a numerous applications and configurations
- Use of OpenCL reduces co-processor integration effort in legacy codes (TPU can only run with Tensorflow)
- Cluster configuration enables larger models to be evaluated by distributing layers to multiple devices
- TRIP's reconfigurability allows hardware to adapt to the applications, maximizing utilizing of available compute resources
- Adding support for sparse matrixes will enable larger datasets to be stored on-chip through Deep Compression
- For extremely large datasets, multi FPGA implementations can be used to provide the required capacity. Data transfer between FPGAs will be chip-chip.
- TRIP is not constrained to a certain number of quantized bits. Based on the application, the size of weights can be reduced to further increase the size of Processing Core without significantly impacting accuracy
- Since TRIP implemented using Off-The-Shelf FPGAs, new technology can be employed by changing design parameters and compiling for the new device(as opposed to spinning new silicon for ASICs)

References

- [1] N. P. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal, R. Bajwa, S. Bates, S. Bhatia, N. Boden, A. Borchers et al., "In-datacenter performance analysis of a tensor processing unit," arXiv preprint arXiv:1704.04760, 2017.
- [2] D. A. Patterson, "Latency lags bandwith," Communications of the ACM, vol. 47, no. 10, pp. 71–75, 2004.
- [3] S. Han, X. Liu, H. Mao, J. Pu, A. Pedram, M. A. Horowitz, and W. J. Dally, "Eie: efficient inference engine on compressed deep neural network," in Proceedings of the 43rd International Symposium on Computer Architecture. IEEE Press, 2016, pp. 243–254.
- [4] A. D. George, M. C. Herbordt, H. Lam, A. G. Lawande, J. Sheng, and C. Yang, "Novo-g#: Large-scale reconfigurable computing with direct and programmable interconnects," in High Performance Extreme Computing Conference (HPEC), 2016 IEEE. IEEE, 2016, pp. 1–7.

Argonne NATIONAL LABORATORY

U.S. DEPARTMENT OF ENERGY

BOSTON UNIVERSITY

Acknowledgments This research used resources of the Argonne Leadership Computing Facility at Argonne National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under contract DE-AC02-06CH11357.