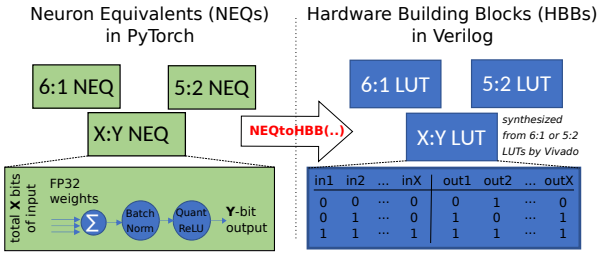
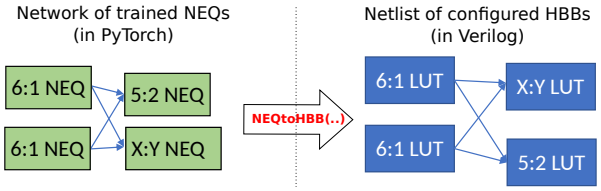
High-Throughput DNN Inference with LogicNets

Yaman Umuroglu, Yash Akhauri, Nicholas J. Fraser and Michaela Blott
Xilinx Research Labs
Dublin, Ireland
Email: {yamanu, yakhauri, nfraser, mblott}@xilinx.com



(a) HBBs and NEQs



(b) Training and converting a network of NEQs to a netlist

Fig. 1. Summary of the LogicNets design flow.

I. Extended Abstract

Deep Neural Networks (DNNs) have a wide application scope beyond computer vision tasks, promising to replace manual algorithmic implementations in applications ranging from large-scale physics experiments to next-generation network security. Such applications may require data processing rates in the millions of samples per second and sub-microsecond latency, which is possible with customized FPGA or ASIC implementations. We present a novel method called LogicNets for co-design of DNN topologies and hardware circuits that maps to a very efficient FPGA implementation to address the needs of such applications.

It is possible to convert a neuron with quantized inputs and outputs into a lookup table (LUT) by evaluating all input-output combinations, as shown in prior work by Nazemi et al. [1]. However, for a neuron with γ inputs of β-bits each, the LUT has $2^{\gamma \cdot \beta}$ entries and this technique cannot be applied to most existing DNNs due to dense connectivity and high activation bitwidth. Instead of mapping existing neural networks to LUTs, we propose to co-design DNN topologies in a way that avoids intractibly large LUTs. The key to designing such DNN topologies is to keep the connectivity γ and activation bitwidth β small. Figure 1 captures the key parts of the LogicNets approach. We first define a set of Neuron Equivalents (NEQs) in PyTorch that map to Hardware Building Blocks (HBBs) generalized as X-input Y-output LUTs. Each NEQ is constrained to have a small number of inputs, such that $X = \gamma \cdot \beta$ maps to at most tens or hundreds of 6:1 FPGA LUTs. Subsequently, we define and train sparsely-connected neural networks in PyTorch and convert the trained networks to netlists in Verilog for synthesis, place and route with Vivado to produce an FPGA bitfile. By exposing the DNN as a netlist, we are also able to exploit synthesis optimizations in existing EDA tools to further compress the DNN to use fewer resources.

TABLE I
Highlights from LogicNets results on the chosen tasks.

Name	Topology	β	γ	Accuracy	LUT	$F_{\max}$
JSC1	4-layer FC	2	3	84.36%	185	1,529 MHz
JSC2	4-layer FC	3	4	87.22%	12,691	471 MHz
JSC3	5-layer FC	3	4	90.88%	34,740	383 MHz
NID1	2-layer FC	2	7	83.88%	3,586	811 MHz
NID2	4-layer FC	3	5	88.44%	12,162	586 MHz
NID3	4-layer FC	2	7	91.43%	27,129	475 MHz

JSC accuracy metric is average area under RoC curve for all classes.

We evaluate our approach on two tasks with very high intrinsic throughput requirements. The first task is the Jet Substructure Classification (JSC) task from [2], part of the L1 trigger at the CERN LHC. The second task is Network Intrusion Detection (NID) by classification of malicious network packets from [3]. We report out-of-context synthesis results targeting a Xilinx `xcvu9p-flgb2104-2-i` FPGA. Our results in Table I indicate that the combination of sparsity and low-bit activation quantization can yield high-speed circuits with small logic depth, low LUT cost and competitive accuracy with throughput in the hundreds of millions of samples per second.

References

[1] M. Nazemi, G. Pasandi, and M. Pedram, "NullaNet: training deep neural networks for reduced-memory-access inference," *arXiv preprint arXiv:1807.08716*, 2018.

[2] J. Duarte, S. Han, P. Harris, S. Jindariani, E. Kreinar, B. Kreis, J. Ngadiuba, M. Pierini, R. Rivera, N. Tran *et al.*, "Fast inference of deep neural networks in FPGAs for particle physics," *Journal of Instrumentation*, vol. 13, no. 07, p. P07027, 2018.

[3] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*. IEEE, 2015, pp. 1–6.

2576-2621/20/$31.00 ©2020 IEEE
DOI 10.1109/FCCM48280.2020.00071