SPARCNet: A Hardware Accelerator for Efficient Deployment of Sparse Convolutional Networks

Tinoosh Mohsenin
CSEE Department, EEHPC Lab, University of Maryland Baltimore County

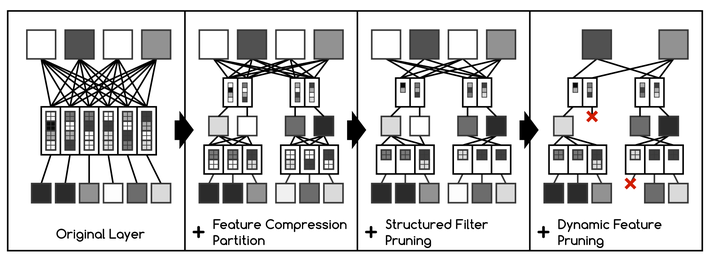
Abstract

Modern convolutional neural networks are very deep and impose significant complexity that is often not feasible in resource-bound, real-time systems that have strict power & area budgets. This work addresses this issue in two key enterprises: targeting efficient sparse-based networks and efficient deployment onto hardware using proposed SPARCNet accelerator. We demonstrate that by targeting the complexity at both the software and hardware level, CNNs can be deployed in resource-bound, real-time settings. When deployed on a Zynq FPGA platform, the reduction techniques enabled up to 6x improvement in efficiency. Compared to CPU counterpart, the SPARCNet accelerator improved throughput by up to 22x while decreasing energy consumption by 13x. The SPARCNet accelerator is further evaluated against a number of other platforms including NVIDIA Jetson TK1 containing K1 GPU. When evaluated on AlexNet, the SPARCNet accelerator running on Zedboard with Zynq FPGA is able to achieve an efficiency of 8.07 GOP/J while under 3 Watts versus Jetson TK1 that obtained an efficiency of 4.58 GOP/J with total system power of 12 Watts.

Reduction Techniques

This work primarily targets the following 3 sparse-based reduction techniques that aim to dissolve away the dense connectivity that is often found at different levels in convolutional neural networks:

Feature Compression Partition (FCP): Introduce branches with each subnetwork containing initial 1x1 conv. compression layer to reduce feature maps (Pre-training).
Structured Filter Pruning (SFP): Statically prune least salient filters in feature maps (post-training).
Dynamic Feature Pruning (DFP): Dynamically prune least salient feature maps for given input (deployment).

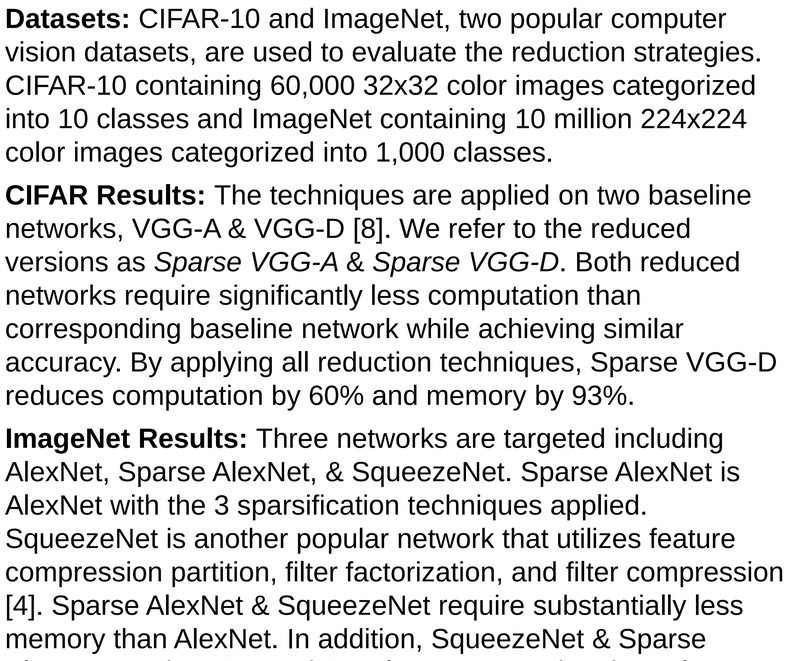


Reduction Evaluation

Datasets: CIFAR-10 and ImageNet, two popular computer vision datasets, are used to evaluate the reduction strategies. CIFAR-10 containing 60,000 32x32 color images categorized into 10 classes and ImageNet containing 10 million 224x224 color images categorized into 1,000 classes.

CIFAR Results: The techniques are applied on two baseline networks, VGG-A & VGG-D [8]. We refer to the reduced versions as *Sparse VGG-A* & *Sparse VGG-D*. Both reduced networks require significantly less computation than corresponding baseline network while achieving similar accuracy. By applying all reduction techniques, Sparse VGG-D reduces computation by 60% and memory by 93%.

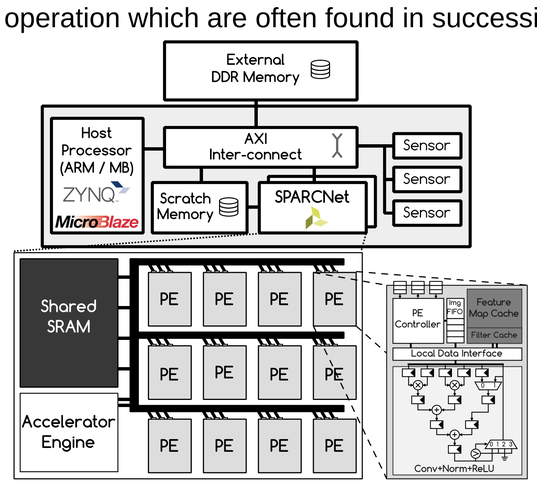
ImageNet Results: Three networks are targeted including AlexNet, Sparse AlexNet, & SqueezeNet. Sparse AlexNet is AlexNet with the 3 sparsification techniques applied. SqueezeNet is another popular network that utilizes feature compression partition, filter factorization, and filter compression [4]. Sparse AlexNet & SqueezeNet require substantially less memory than AlexNet. In addition, SqueezeNet & Sparse AlexNet require 49% and 21% less computation than AlexNet.



SPARCNet Accelerator

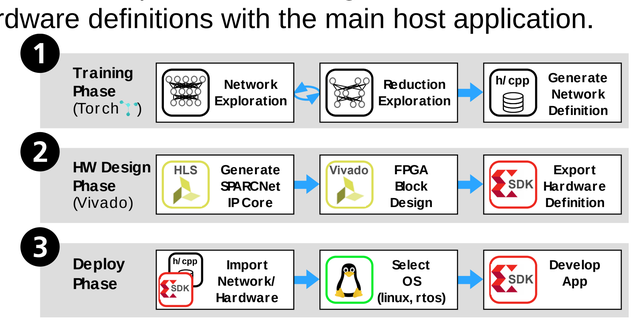
The key objective is to efficiently deploy CNNs in resource-bound, real-time settings by efficiently accelerating convolutional layers. The main FPGA components consist of a host processor, SPARCNet accelerator, memory, and AXI interconnect backbone. The processing engines (PEs) are arranged in a grid configuration with a unified network interface. The three key innovations of the accelerator include:

Efficient Parallelization: PEs parallelize primarily across output channels/feature maps. This tiling method is most efficient of the three tiling methods- requiring no inter-PE communication and lowest memory bandwidth [14].
Sparsification Support: Support for the three proposed sparsification techniques by containing both a filter and input channel bit mask to define connectivity between input channels and feature maps.
Fused Operations: Support for fusion of convolution, batch normalization, and rectified linear unit (ReLU) layers into a single operation which are often found in succession.



SPARCNet Framework

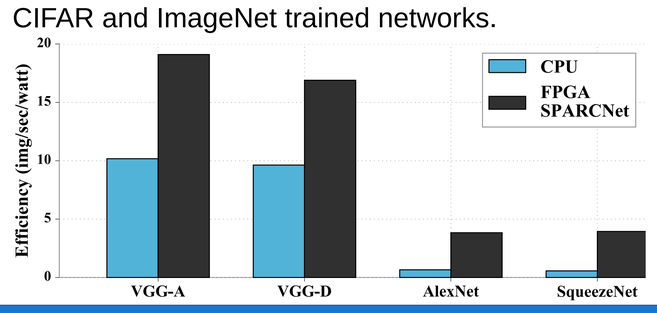
A companion framework was also designed to help with each of the three main phases of development. The first phase corresponds to training which is used to explore different network topologies and reduction techniques along with generation of optimized network definition. The second phase is targeted towards developing the hardware definition which is done using an Xilinx toolchain flow. This involves customizing the SPARCNet IP core to the specific network definition such as the number of PEs, scratchpad memory and weight format. The SPARCNet IP is integrated and synthesized with the rest of the FPGA fabric and exported as a hardware definition. The third and final phase is the integration of the network and hardware definitions with the main host application.

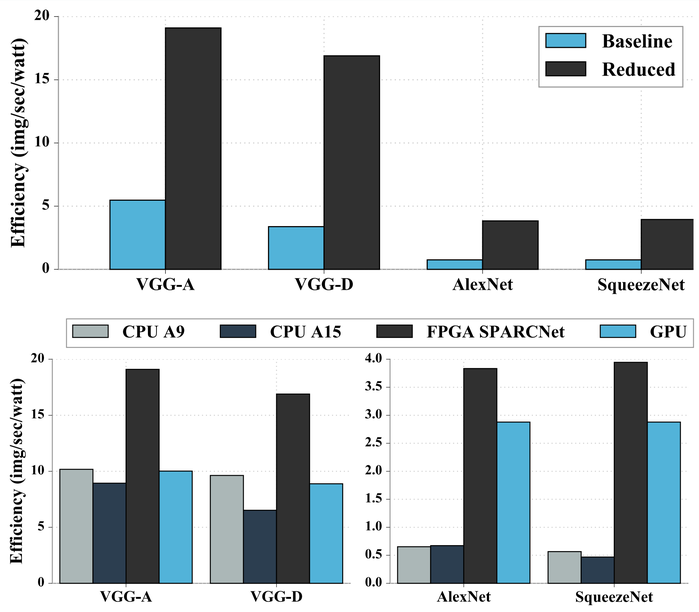


Evaluation on COTS

The proposed accelerator was evaluated when running in real-time on a number of FPGA-based SoC platforms for each network. The FPGA platforms include the Zedboard with Zynq SoC containing dual ARM A9 as well as Arty/Cmod-A7 containing Artix-7 35 FPGA with MicroBlaze soft-processor. For each platform, the corresponding CPU is used as a baseline reference. Furthermore, we alternatively target NVIDIAs low-power Jetson TK1 platform with K1 SoC containing quad ARM A15 & K1 GPU. The primary metric used for comparison is energy efficiency (img/sec/watt).

1. **CPU vs FPGA:** The accelerator is able to increase efficiency by 2x for VGG-A & VGG-D and by 6.5x for AlexNet & SqueezeNet compared to the host CPU.
2. **Baseline vs Reduced:** By exploiting the reduction techniques and SPARCNet hardware support, efficiency can be increased by 4-6x using the reduced networks.
3. **CPU vs FPGA/GPU:** Both the FPGA-based SPARCNet accelerator and GPU improve efficiency over standalone CPU. Furthermore, the SPARCNet accelerator is able to achieve 90% and 35% improvement over the GPU for CIFAR and ImageNet trained networks.





Comparison with Prior Work

The SPARCNet accelerator is compared against existing FPGA accelerators. Since prior works evaluate on different networks, we normalize based on the number of operations required by the network. When deployed on Zedboard platform with dual ARM A9 running at 667.67 MHz & FPGA running at 100 MHz, the proposed accelerator achieves an energy efficiency of 8.07 GOP/J with total system power of 2.79 W and throughput of 22.31 GOP/s. This achieves a much higher efficiency than the prior best accelerators [14] and [10] while targeting a much lower power utilization.

Metrics	[2]	[14]	[10]	**This Work**	
Platform	Zynq-7 XC7Z045	Virtex-7 VX485T	Stratix V GSD8	**Zynq-7 XC7Z020**	Jetson TK1
Precision	16-bit Fixed	32-bit Float	8/16-bit Fixed	**16-bit Float**	32-bit Float
Clock (MHz)	150	100	120	**100**	2320
Network	NA	AlexNet	VGG-16	**AlexNet**	AlexNet
Complexity (GOP)	0.552	2.15	30.9	**2.15**	2.15
Performance (GOP/s)	23.18	61.62	117.8	**22.31**	55.30
Total Power (W)	8	18.61	25.8	**2.79**	12.08
Energy Efficiency (GOP/J)	2.90	3.31	4.57	**8.07**	4.58

Conclusion

This work proposed contributions in two enterprises to enable deploying CNNs in resource-bound, real-time settings. The first demonstrated the effectiveness of the proposed sparsification techniques when evaluated on several networks. The second demonstrated the SPARCNet accelerator and framework when evaluated in real-time on different FPGA platforms. The accelerator is able to outperform both embedded GPU and FPGA-based accelerators. When accelerating AlexNet on Zedboard platform, the proposed accelerator achieves an energy efficiency of 8.07 GOP/J with total system power of 2.79 W and throughput of 22.31 GOP/s. This achieves higher efficiency than the prior best accelerators as well Jetson TK1 while targeting a power profile that is more than 4x lower than the Jetson TK1. When accelerated on a Zynq FPGA platform, the reduction techniques enabled up to 6x improvement in energy efficiency over the baseline network. Furthermore, relative to the host CPU, SPARCNet accelerator improves throughput by up to 22x while decreasing energy by 13x.

References

[1] Y.-H. Chen et al. Eyeriss: An Energy-efficient Reconfigurable Accelerator for Deep Convolutional Neural Networks. In 2016 IEEE International Solid-State Circuits Conference (ISSCC), pg. 262–263. IEEE, 2016.
[2] V. Gokhale, J. Jin, A. Dundar, B. Martini, and E. Culurciello. A 240 G-ops/S Mobile Coprocessor For Deep Neural Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 682–687, 2014.
[3] K. He, X. Zhang et al. Deep Residual Learning for Image Recognition. CoRR, abs/1512.03385, 2015.
[4] F. N. Iandola et al. Squeezenet: Alexnet-level Accuracy with 50x Fewer Parameters and <1MB Model Size. CoRR, abs/1602.07360, 2016.
[5] X. Liu et al. Reno: A High-Efficient Reconfigurable Neuromorphic Computing Accelerator Design. In Proceedings of the 52Nd Annual Design Automation Conference, DAC '15, pages 66:1–66:6, New York, NY, USA, 2015. ACM.
[6] A. Page, A. Jafari, C. Shea, and T. Mohsenin. Sparcnet: A Hardware Accelerator For Efficient Deployment of Sparse Convolutional Networks. Journal on Emerging Technologies in Computing (JETC).
[7] J. Sim et al. A 1.42 TOPS/W Deep Convolutional Neural Network Recognition Processor for Intelligent IOE Systems. In 2016 IEEE International Solid-State Circuits Conference (ISSCC), pages 264–265. IEEE, 2016.
[8] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-zdcale Image Recognition. CoRR, abs/1409.1556, 2014.
[9] L. Song and others. C-Brain: A Deep Learning Accelerator that Tames the Diversity Of CNNs Through Adaptive Data-level Parallelization. In 2016 53nd ACM/EDAC/IEEE Design Automation Conference (DAC), pages 1–6, June 2016.
[10] N. Suda et al. Throughput-Optimized OpenCL-based FPGA Accelerator For Large-scale Convolutional Neural Networks. In Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, FPGA '16, pgs16–25, New York, NY, USA, 2016. ACM.
[11] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going Deeper With Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1–9, 2015.
[12] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the Inception Architecture for Computer Vision. arXiv preprint arXiv:1512.00567, 2015.
[13] Y. Wang et al. DeepBurning: Automatic Generation of FPGA-Based Learning Accelerators for the Neural Network Family. In 2016 53nd ACM/EDAC/IEEE Design Automation Conference (DAC), pg. 1-6, 2016.
[14] C. Zhang et al. Optimizing FPGA-based Accelerator Design For Deep Convolutional Neural Networks. In Proceedings of the 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, pages 161–170. ACM, 2015.