Aggressive Compression of MobileNets Using Hybrid Ternary Layers

Dibakar Gope, Jesse Beu, Urmish Thakker, and Matthew Mattina
Arm ML Research Lab

Problem to be solved: In a neural network with binary (-1, 1) or ternary (-1, 0, 1) weights, multiplications are replaced by additions. Multipliers consume significantly more area and energy than adders. A specialized hardware, such as ASICs and FPGAs can therefore accommodate considerably more adders in place of multipliers, potentially achieving both higher throughput and significant savings in per-op energy for a neural network with binary or ternary weights. However, state-of-the-art approaches to binary and ternary quantization either drop prediction accuracy significantly for MobileNets [1] (as shown by ternary weight networks (TWN) [2] in TABLE I.) or cause a prohibitive (317.5%) increase in additions to preserve the baseline accuracy, degrading the throughput of neural network inference (as shown by StrassenNets [3] in TABLE I.).

Technical approach and its novelty: For every DNN layer, StrassenNets casts the (matrix) multiplication of weight matrix with activations as a 2-layer sum-product network (SPN). The exorbitant increase in additions stems from the use of a large number of hidden units in the SPNs needed to closely approximate each convolutional filter in a network layer. While this might be required for some of the convolutional filters in a layer, our observations indicate that not all filters require wide hidden layers. As different filters in a layer tend to capture different features, some being more complicated than others, they respond differently to ternary quantization, and, in turn, to *strassenified* convolution at varied hidden layer widths. Furthermore, due to the hidden unit reuse in SPN, a group of filters with sub-filter similarities may respond more favorably to ternary quantization than outlier filters within the same layer extracting significantly different features. Guided by these insights, we propose a layer-wise hybrid filter banks for MobileNets that performs traditional convolutions for precision critical filters with full-precision weight values, whereas ternary quantization tolerant filters perform strassenified convolutions using narrow hidden layers. See [4] for details. Recent works [5, 6, 7, 8] have exploited other forms of hybridization techniques to capture low- and high-fidelity features and in turn to compress state-of-the-art CNN and RNN networks.

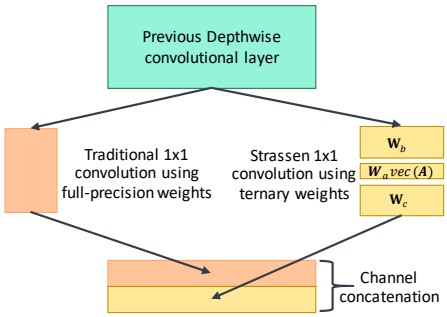

Fig. 1. A MobileNets pointwise layer with hybrid filter bank.

TABLE I. Performance of MobileNet-V1 (width multiplier 0.5) over ImageNet dataset on an area-equivalent hardware accelerator

Network	Accuracy (Top-1)	Model size (KB)	Energy/inf. (normalized)	Throughput (normalized)
Baseline full-precision	65.2%	2590	1	1
TWN	55.54%	323.7	0.2	2
StrassenNets	65.14%	1178.9	0.9	0.46
Hybrid filter bank	64.69%	1267.1	0.72	1

Results: As shown in TABLE I., the hybrid filter banks achieve 46.4% and 51.07% reduction in multiplications and model size respectively while incurring modest (48%) increase in additions. This translates into 28% savings in energy required per inference while ensuring no degradation in throughput on a DNN hardware accelerator consisting of both MAC and adders when compared to the execution of baseline MobileNets on a MAC-only hardware accelerator.

Significance for the tinyML community: MobileNets family of computer vision neural networks have fueled tremendous progress in the design and organization of resource-efficient architectures in recent years. New applications with stringent real-time requirements on highly constrained devices require further compression of MobileNets to make it amenable for edge devices. The hybrid filter bank is a first step towards ternarizing the already compute-efficient MobileNets with a negligible loss in accuracy on a large-scale dataset such as ImageNet, better enabling deployment for vision-based “tinyML” applications. See [4] for details.

REFERENCES

[1] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. CoRR, abs/1704.04861, 2017.

[2] F. Li, and B. Liu, "Ternary weight networks," 2016.

[3] M. Tschannen, A. Khanna, and A. Anandkumar, "StrassenNets: Deep learning with a multiplication budget," in ICML, 2018.

[4] Dibakar Gope, Jesse Beu, Urmish Thakker, and Matthew Mattina. Ternary MobileNets via Per-Layer Hybrid Filter Banks. CoRR, abs/1911.01028, 2019.

[5] Dibakar Gope, Ganesh Dasika, and Matthew Mattina. Ternary hybrid neural-tree networks for highly constrained iot applications. CoRR, abs/1903.01531, 2019.

[6] Urmish Thakker, Jesse G. Beu, Dibakar Gope, Ganesh Dasika, and Matthew Mattina. Run-time efficient RNN compression for inference on edge devices. CoRR, abs/1906.04886, 2019.

[7] Urmish Thakker, Jesse G. Beu, Dibakar Gope, Chu Zhou, Igor Fedorov, Ganesh Dasika, and Matthew Mattina. Compressing rnns for iot devices by 15-38x using kronecker products. CoRR, abs/1906.02876, 2019.

[8] Urmish Thakker, Igor Fedorov, Jesse Beu, Dibakar Gope, Chu Zhou, Ganesh Dasika, and Matthew Mattina. Pushing the limits of rnn compression. CoRR, abs/1910.02558, 2019.