A. Channel Scaling Vs. Feature map Scaling

For extremely lower ReLU budgets, we use a combination of channel scaling and fmaps' resolution scaling. Since reducing fmaps' resolution (each spatial dimensions of fmaps) by 2× (ρ=0.5) decreases the ReLU count by 4×, we first use channel scaling (α = 0.5) for reducing the ReLU count by 2×. Further, for 4× reduction in ReLU count, we prefer to use fmaps' resolution scaling (ρ=0.5) over channel scaling (α=0.25) since the former results in more accurate networks, as illustrated in Table 9. Unlike fmap resolution scaling, channel scaling reduces the parameter count along with the ReLU count, which may reduce the expressive power of a network. Hence, the network is more accurate with fmaps' resolution scaling.

Table 9. Performance comparison for ReLU optimization using channel scaling (α <1) and fmap-resolution scaling (ρ <1). Baseline models have α=1 and ρ=1. At iso-ReLU, accuracy (w/ KD) of the the fmap-resolution scaled models is higher than the channel scaled models.

Network	#Conv	#ReLUs	CIFAR-100	
			W/o KD (%)	W/ KD (%)
ResNet18 (baseline)	17	557K	74.46	76.94
ResNet18; α=0.25, ρ=1	17	139K	68.17	70.19
ResNet18; α=1, ρ=0.5	17	139K	68.47	72.72
ResNet10 (baseline)	9	311K	74.10	76.69
ResNet10; α=0.25, ρ=1	9	78K	66.69	66.88
ResNet10; α=1, ρ=0.5	9	78K	66.67	71.86
ResNet6 (baseline)	5	188K	68.86	69.58
ResNet6; α=0.25, ρ=1	5	47K	57.64	56.9
ResNet6; α=1, ρ=0.5	5	47K	64.74	68.09

B. VGGNet DeepReDuce Pareto Points

We do not remove ReLUs from fully connected (FC) layers as FC account only 8.192K ReLUs and training networks without FC ReLUs is challenging. The results are shown in Table 10. Unlike ResNets and MobileNets, ReLUs in S_5 are least critical and that of the S_1 is moderate.

Table 10. Stage-wise ReLUs' criticality in VGG16 evaluated on CIFAR-10. S_5 is least critical while S_2 and S_3 are most critical.

Net	#ReLUs	W/o KD (%)	W/ KD (%)	C_k
S_1 + FC-ReLUs	139.3K	82.0	81.4	10.90
S_2 + FC-ReLUs	73.7K	86.1	85.3	14.31
S_3 + FC-ReLUs	57.3K	86.4	85.1	14.40
S_4 + FC-ReLUs	32.8K	77.1	77.7	9.17
S_5 + FC-ReLUs	14.3K	63.9	66.0	0.00

The ReLU optimizations step for the Pareto points in Figure 5 are listed in Table 11. Models are the ReLU-optimized versions (Thinned and Reshaped) of two Culled networks: (1) stages S_4 and S_5 are Culled and (2) stages S_1, S_4 and S_5 are Culled.

Table 11. Optimization steps for MobileNetV1 DeepReDuce models shown in Figure 5

Optimization Steps	#ReLUs	Acc.(%)
S_1 + S_2 + S_3 + FC	253.95K	93.92
S_2 + S_3 + FC	122.88K	92.52
S_2^{RT} + S_3^{RT} + FC	73.73K	90.23
S_1^{RT} + S_2^{RT} + S_3^{RT} + FC, α=0.5	69.63K	89.97
S_2^{RT} + S_3^{RT} + FC, α=0.5	36.86K	88.92

C. ReLUs' Criticality and Pareto Points for MobileNets

We evaluate the ReLUs' criticality in MobileNetV1 (Howard et al., 2017) and MobileNetV2 (Sandler et al., 2018)) on the CIFAR-100. The results are shown in Table 12. We observed the similar trend as ResNet18 and ResNet34 on CIFAR-100/TinyImageNet (shown in Tables 1 and 3), accuracy differs significantly across stages and S_1 (S_4) ReLUs are least (most) critical.

Table 12. Stage-wise criticality of ReLUs in MobileNetV1 and MobileNetV2 evaluated on CIFAR-100. FR is baseline with Full-ReLU (S_1+S_2+S_3+S_4+S_5). ReLUs in S_1 (S_4) are least (most) critical.

Net	MobileNetV1				MobileNetV2			
	#ReLUs	W/o KD(%)	W/ KD(%)	C_k	#ReLUs	W/o KD(%)	W/ KD(%)	C_k
FR	411.6K	67.58	-	-	425.6K	68.46	-	-
S_1	131.1K	33.06	34.16	0.00	196.6K	37.82	34.25	0.00
S_2	114.7K	49.64	50.65	11.83	110.6K	49.83	46.93	9.12
S_3	57.3K	55.56	54.20	15.09	58.4K	54.74	53.06	14.15
S_4	94.2K	57.37	61.10	19.60	27.6K	57.08	57.28	18.26
S_5	14.3K	42.32	45.45	9.37	32.4K	48.42	50.49	12.73

The Pareto points of DeepReDuce models for MobileNetV1 (CIFAR-100) are shown in Figure 6. The optimization steps for all DeepReDuce models are list in the Table 13.

First, in step 1 of DeepReDuce (Figure 4), we Culled the least critical stage S_1. In step 2 of ReLU Thinning, we had two ways to remove the ReLUs from alternate layers, either from 3×3 depthwise convolution layer or 1×1 pointwise convolution layer. When downsampling is performed in 3×3 depthwise convolution layer, the ReLU count of both the layers are not equal. More precisely, #ReLUs in the 1×1 pointwise convolution is twice as that in the preceding 3×3 depthwise conv.

We empirically found that removing ReLUs from 3×3 depthwise conv layer yields more accurate iso-ReLU models. We suspect this is because 3×3 depthwise convolutions perform filtering (feature learning) and 1×1 pointwise convolutions perform feature aggregation (Howard et al., 2017), the ReLUs in the former layer is more critical for accuracy.

Table 13. Optimization steps for MobileNetV1 DeepReDuce models shown in Figure 6

Optimization Steps	#ReLUs	Acc.(%)
S_2 + S_3 + S_4 + S_5	280.60K	70.83
S_2^{RT} + S_3^{RT} + S_4^{RT} + S_5^{RT}	108.54K	70.77
S_2^{RT} + S_3^{RT} + S_4^{RT} + S_5^{RT}, α=0.5	54.27K	67.46
S_2^{RT} + S_3^{RT} + S_4^{RT} + S_5^{RT}, ρ=0.5	27.14K	62.96
S_2^{RT} + S_3^{RT} + S_4^{RT} + S_5^{RT}, α=0.5, ρ=0.5	13.57K	58.25

D. ReLU Criticality in ResNet56

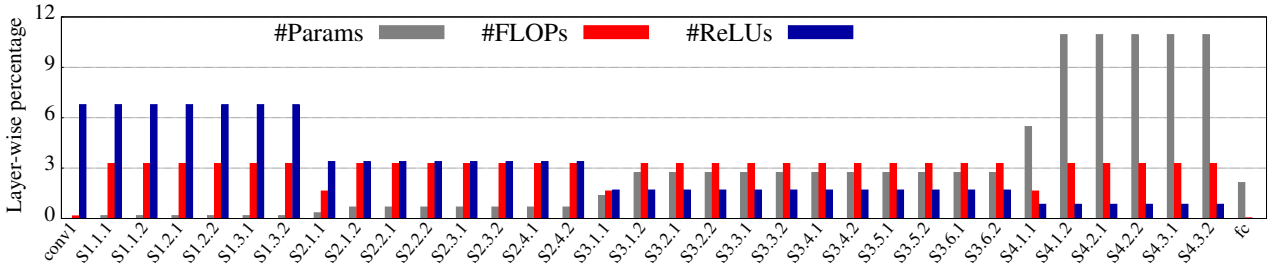
We examine the stage-wise criticality of ReLUs in ResNet56 and results are shown in Table 14.

Table 14. Stage-wise criticality of ReLUs in ResNet56 evaluated on CIFAR-100. $S3$ is most critical and $S1$ is least critical.

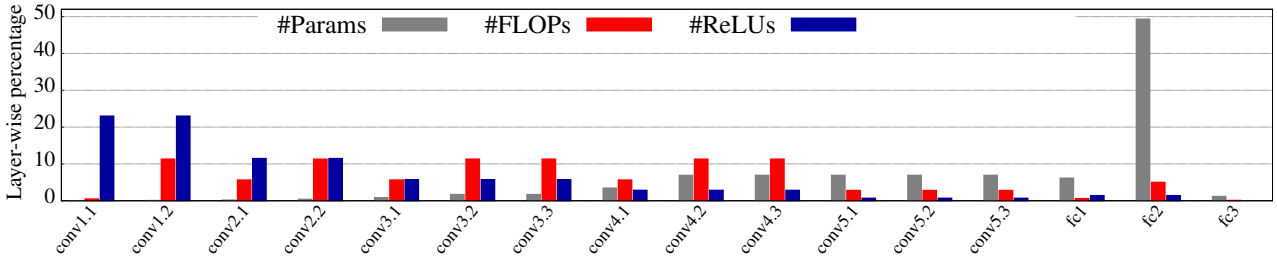
Stages	#ReLUs	W/o KD (%)	W/ KD (%)	C_k
S1	311.3K	57.92	59.45	0.0
S2	147.5K	65.62	67.97	6.0
S3	73.73K	65.36	69.22	7.2

E. Layer-wise Distribution of ReLUs

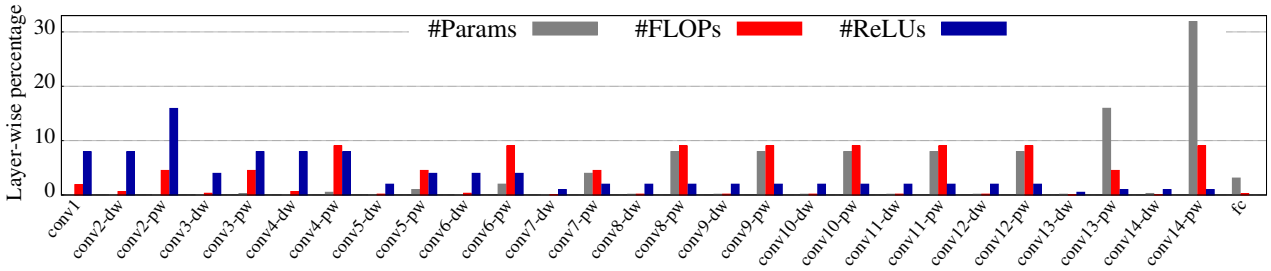
We show the layer-wise distribution of FLOPs, parameters, and ReLU count in the standard networks such as ResNet34 (He et al., 2016), VGG16 (Simonyan & Zisserman, 2014), MobileNetV1 (Howard et al., 2017), and MobileNetV2 (Sandler et al., 2018) in Figure 7. Consistent with ResNet18 (Figure 3), the FLOPs are evenly distributed, more parameters are used in deeper layers, and ReLUs are mostly in initial layers of the networks. Thus, the ReLU reduction in initial layers has a significantly greater impact on the overall ReLU count of these networks. Moreover, the stark difference between the ReLU distribution and FLOPs/parameter distribution indicates that ReLU optimization cannot be ensured through the popular FLOPs/parameters pruning techniques.



(a) Layer-wise distribution of parameter, FLOPs, and ReLUs in ResNet34 (He et al., 2016).



(b) Layer-wise distribution of parameter, FLOPs, and ReLUs in VGG16 (Simonyan & Zisserman, 2014).



(c) Layer-wise distribution of parameter, FLOPs, and ReLUs in MobileNetV1 (Howard et al., 2017).

Figure 7. Layer-wise percentage of parameter, FLOPs, and ReLUs in various DNNs. FLOPs are evenly distributed, parameters (ReLUs) are increases (decreases) from initial to deeper layers.