Cascaded channel pruning using hierarchical self-distillation

Roy Miles
r.miles18@imperial.ac.uk
Krystian Mikolajczyk
k.mikolajczyk@imperial.ac.uk

Imperial College London
Department of Electrical and Electronic Engineering
London, UK

1 ResNet50 performance metrics

Table 1 shows the complexity and parameter break-down for each layer in the ResNet-50 model with an input image of dimensions $224 \times 224 \times 3$.

Block #	$w \times h$	#Filters	FLOPs	Params
	224×224	64	118M	0.01M
0	56×56	$[64, 64, 256]$	231M	0.07M
1	56×56	$[64, 64, 256]$	218M	0.07M
2	56×56	$[64, 64, 256]$	218M	0.07M
3	56×56	$[128, 128, 512]$	295M	0.38M
4	28×28	$[128, 128, 512]$	218M	0.28M
5	28×28	$[128, 128, 512]$	218M	0.28M
6	28×28	$[128, 128, 512]$	218M	0.28M
7	28×28	$[256, 256, 1024]$	295M	1.51M
8	14×14	$[256, 256, 1024]$	218M	1.11M
9	14×14	$[256, 256, 1024]$	218M	1.11M
10	14×14	$[256, 256, 1024]$	218M	1.11M
11	14×14	$[256, 256, 1024]$	218M	1.11M
12	14×14	$[256, 256, 1024]$	218M	1.11M
13	14×14	$[512, 512, 2048]$	295M	6.03M
14	7×7	$[512, 512, 2048]$	218M	4.46M
15	7×7	$[512, 512, 2048]$	218M	4.46M
	1×1	1000	2.05M	2.05M
Total:			3.85B	25.5M

Table 1: Performance statistics for the ResNet50 architecture on the ImageNet2012 dataset.

2 Layer-wise pruning

Figure 1 shows the percentage of pruned filters in each layer for a student model and two TAs. The student uniformly prunes the layers, while the larger TA models focus on the last layers. This is in contrast to how most other pruning methodologies work, which tend to

© 2020. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

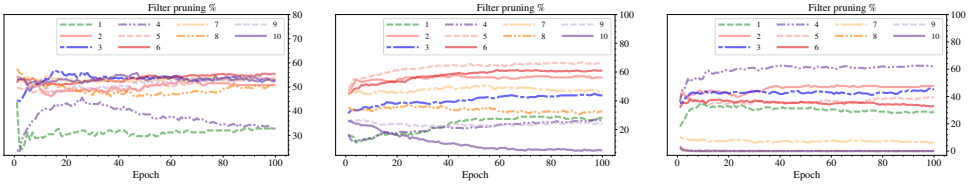

Figure 1: The layer-wise pruning for a student and two TA models trained using cascaded pruning. From left to right are models T_0, T_1, and T_2 respectively. Each TA uses the VGG16 architecture and is jointly trained on the CIFAR10 dataset.

Layer #	$w \times h$	#Filters	FLOPs	Params	Filter pruning			
					$k_0 = 0.1$	$k_0 = 0.5$	$k_0 = 0.6$	$k_0 = 0.8$
0	32×32	64	1.77M	1.73K	0%	0%	0%	0%
1	32×32	64	37.75M	36.86K	8.5%	20.1%	50.8%	89.4%
2	16×16	128	18.87M	73.73K	1.4%	11.7%	29.6%	83.2%
3	16×16	128	37.75M	0.15M	40.0%	45.4%	51.1%	67.7%
4	8×8	256	18.87M	0.29M	17.8%	33.3%	45.3%	63.1%
5	8×8	256	37.75M	0.59M	11.1%	36.0%	46.1%	77.2%
6	8×8	256	37.75M	0.59M	37.3%	59.0%	61.2%	86.6%
7	4×4	512	18.87M	1.18M	30.1%	60.8%	61.5%	73.7%
8	4×4	512	37.75M	2.36M	0%	29.7%	65.9%	67.3%
9	4×4	512	37.75M	2.36M	0%	17.0%	56.5%	92.3%
10	2×2	512	9.44M	2.36M	6.0%	69.0%	75.9%	88.9%
11	2×2	512	9.44M	2.36M	0%	85.5%	72.4%	81.7%
12	2×2	512	9.44M	2.36M	0%	63.5%	63.7%	84.1%
13	1×1	512	0.26M	0.26M	0%	31.8%	31.9%	42.1%
14	1×1	512	5.12K	5.12K	0%	0%	0%	0%

Table 2: Pruning % in each layer as a result of cascaded pruning on the CIFAR10 dataset and with the VGG16 architecture at varying filter-pruning ratios. The last two layers (13 & 14) are the two dense classification layers which are not masked.

result in significant pruning for the last few layers of the network. Table 2 shows how this distribution of pruning levels changes with the filter pruning ratio.

2.1 Uniformly pruned baselines and KD loss terms

The empirical results demonstrated by [1] showed that most channel pruning pipelines achieve comparable or worse performance to training the equivalent smaller model from scratch. Therefore, to confirm the performance benefits of our proposed method, we compare our results against individually training two smaller VGG16 variants from random initialisation. Specifically, we consider using both width scaling and shuffle units [2]. Width scaling reduces the depth of each layer by a given %, while a shuffle unit replaces the convolutional layers with group convolutions and channel shuffles. We use the same training methodology as the original baseline for all these models, which lasts 150 epochs with a cosine learning rate schedule. Liu *et al*. [1] considered two training schemes for these uniformly pruned baseline: training for the same number of epochs as the baseline and training for the same computational budget. In both cases, the reported accuracy's were similar, and in our evaluation we found that further training any of these uniformly pruned baseline results had little effect on the accuracy.

To provide a thorough evaluation of cascaded pruning, we also consider the impact of

Method	Params	FLOPs	Top-1 Accuracy (%)
Baseline	14.98M	313M	93.26%
Standard-0.75	8.48M	184M	↓ 5.69%
Standard-0.5	3.82M	89M	↓ 6.82%
Standard-0.25	**1.00M**	**28M**	↓ 11.08%
Group-2	7.62M	158M	↓ 6.46%
Group-4	3.95M	80M	↓ 7.69%
Ours-0.5 None	4.13M	102M	↓ 2.40%
Ours-0.5 w/ KD	3.62M	97M	↓ **0.79**%
Ours-0.5 w/ Hints	3.61M	96M	↓ 2.42%
Ours-0.5 w/ KD & Hints	3.69M	99M	↓ 1.27%

Table 3: Accuracy and performance metrics for two efficient VGG16 variants trained from random initialisation on the CIFAR10 dataset. Group-*g* indicates the use of group convolutions with *g* groups, while Standard-*s* uses *s*% width scaling for all the convolutional layers.

using the explicit KD and hint loss terms between each student-teacher pair. We use only a single TA with a filter pruning ratio of 0.5 and set $\lambda_H = 0.001$ and $\lambda_{KD} = 0.4$ throughout. These complete sets of results can be seen in table 3. In the case where no KD or hinted losses are used, only implicit knowledge is distilled between the models, as attributed to the sharing of weights and joint training of all the models. The KD loss term between each teacher-student pair significantly increases the student's performance, while the hinted losses damage the student's performance. The hinted losses perform poorly in this framework since the enabled filters are constantly changing through the importance score updates. However, the student models trained using cascaded pruning still significantly outperform the equivalent smaller models when trained from scratch. These results demonstrate how the learnt mask structure is an integral part and contributing factor to the performance of these cascaded pruned networks.

References

[1] Zhuang Liu, Mingjie Sun, Tinghui Zhou, Gao Huang, and Trevor Darrell. Rethinking The Value Of Network Pruning. *ICLR*, 2019.

[2] Xiangyu Zhang, Xinyu Zhou, and Mengxiao Lin. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. *CVPR*, 2018.