KNAPSACK PRUNING WITH INNER DISTILLATION: SUPPLEMENTARY MATERIAL

Anonymous authors
Paper under double-blind review

1 PRUNING OF ECA-RESNET-D

In this section, we present results that we obtained by pruning a neural network superior to ResNet50. We choose to prune ECA-ResNet-D. This network has a backbone of ResNet, but we add two modifications. First, we change the stem cell as presented in He et al. (2019). In addition, we add ECA modules, as suggested in Wang et al. (2019). The obtained architecture performs better than classical ResNet, and thus, pruning this network is more interesting. We have pruned different versions of ECA-ResNet-D with our method using two criteria: Flops based pruning, as described in the paper, and Inference-time based pruning. In this setting, we measure the inference time of every convolutional layer of the network, and apply our knapsack method where, instead of imposing a constraint on the total FLOPS of the pruned network, we impose a constraint on the final inference time. Most of the pruning method focus on reducing the total FLOPS of the pruned networks, but this measure does not often reflects the real inference time on a GPU, and networks with few flops such as EfficientNet Tan & Le (2019) runs with a lower throughput than heavier architecture such as ResNet. In the below table, we present the results of pruning several versions of ECA-ResNet-D, with both FLOPS based pruning and time-based pruning.

Network	Accuracy	Speed (Im/sec) on V100	Speed (Im/sec) on P100	Flops (Gigas)
Unpruned Architectures				
Baseline, Resnet50D	79.30%	2728	791	4.35
ECA Resnet-50D	80.61%	2400	718	4.35
ECA Resnet-101D	82.19%	1476	444	8.07
Flops based pruned				
ECA Resnet-50D 41%	80.10%	2434	924	2.56
ECA Resnet-101D 55%	81.24%	1651	642	3.61
ECA Resnet-101D 68%	80.69%	1735	717	2.58
Time based pruned				
ECA Resnet-50D 43%	79.71%	3587	1200	2.53
ECA Resnet-50D 39%	79.89%	3145	1121	2.66
ECA Resnet-50D 21%	80.34%	2653	906	3.45
ECA Resnet-101D 57%	80.86%	2791	1010	3.47

Table 1: Performance of FLOPS based and Time based pruning of ECA-ResNet on Imagenet Dataset

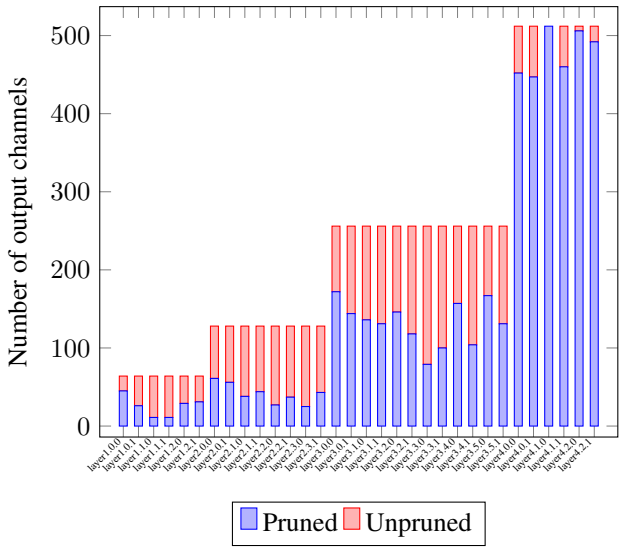
We can see how time-based pruning using our knapsack method provided extremely fast networks. For example, time-pruning of 57% of ECA-ResNet-101D provided a network with 80.86% accuracy while inferring at 1010 images per seconds on a P100 GPU. To the best of our knowledge, this is the fastest network of a NVIDIA P100 GPU to get an accuracy above 80% on ImageNet. The above networks and their checkpoints have been integrated on the famous Ross Wightman repository Wightman (2019), and are available to the public.

2 KNAPSACK PRUNING ALGORITHM

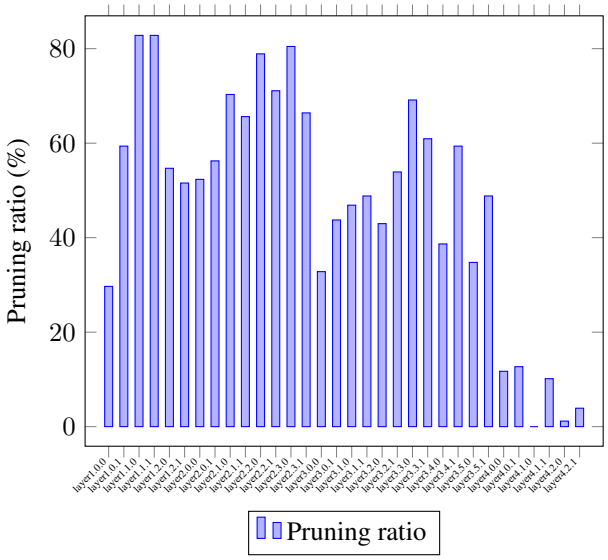
Algorithm 1 Knapsack Pruning

input $C, w_i \ \forall i$
  for all $0 \leq i \leq n$ **do**
    Compute $I_i \leftarrow I(w_i)$
    Compute $F_i \leftarrow F(w_i)$
  end for
  Compute $G \leftarrow \mathrm{GCD}(F_i)$
  for all i **do**
    $F_i \leftarrow F_i/G$
  end for
  $C \leftarrow C/G$
  Initialize $T \leftarrow$ 0-float array of size $2 \times C$
  Initialize $K \leftarrow$ **False**-binary array of size $n \times C$
  for all i **do**
    $I_{\text{curr}} \leftarrow I_i, F_{\text{curr}} \leftarrow F_i$
    $i_{\text{prev}} \leftarrow (i-1)\%2, i_{\text{curr}} \leftarrow i\%2$
    for all $0 \leq f \leq C$ **do**
      if $f \geq F_i$ **then**
        $v_1 \leftarrow I_i + T[i_{\text{prev}}][f - F_i]$
      else
        $v_1 \leftarrow 0$
      end if
      $v_2 \leftarrow T[i_{\text{prev}}][f]$
      if $F_i \leq f$ and $v_2 \leq v_1$ **then**
        $T[i_{\text{curr}}][f] \leftarrow v_1$
        $K[i][f] \leftarrow$ **True**
      else
        $T[i_{\text{curr}}][f] \leftarrow v_2$
      end if
    end for
  end for
  $P \leftarrow []$
  for i from n to 0 decreasing by 1 **do**
    if $K[i][F]$ is **True then**
      $P \leftarrow [P, i]$
      $K \leftarrow K - F_{i-1}$
    end if
  end for
output P

3 PRUNING RATIO AND PRUNED OUTPUT CHANNELS FOR RESNET 50



(a) Number of output channels for ResNet 50



(b) Pruning ratio of output channels for ResNet 50

Figure 1: Pruning ratio and pruned output channels for ResNet 50

REFERENCES

T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li. Bag of tricks for image classification with convolutional neural networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 558–567, 2019.

Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6105–6114, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

Qilong Wang, Banggu Wu, Pengfei Zhu, Peihua Li, Wangmeng Zuo, and Qinglei Hu. Eca-net: Efficient channel attention for deep convolutional neural networks. *ArXiv*, abs/1910.03151, 2019.

Ross Wightman. *PyTorch image models repository, url:https://github.com/rwightman/pytorch-image-models*, 2019. URL https://github.com/rwightman/pytorch-image-models.