EAPruning: Evolutionary Pruning for Vision Transformers and CNNs

Qingyuan Li[1] Bo Zhang[1] Xiangxiang Chu[1]

[1]Meituan

Background

- *Criteria-based Pruning* Structured pruning usually removes redundant filters according to some human-crafted criteria. It usually requires to identify the relative importance of filters. Recent research shows that structure of network is more important than filter weights.
- *Search-based Pruning* Search-based methods consider pruning as an architecture search paradigm. They search an optimal subnet by using RL, GS or NAS. But these methods involve strong domain expertise, require extra hyper-parameter tuning etc.

EAPruning

- *Pruning Space* We use channel-number encoding instead of channel-wise one-hot encoding. This can greatly reduce search space.
- *Channel Selection* We assume that the performance of the subnet only has to do with the structure, we just randomly sample channels to a target number.
- *Weight-Reconstruction* We use the technique of weight reconstruction to avoid performance collapse of subnetworks.
- *Pruning Search* We choose NSGA-III instead of vanilla evolution. Firstly, it can maintain the diversity of the population. Secondly, we can sample multiple subnets that meet different constraints from the Pareto front at the end.

Method

Our whole pipeline can be viewed in Figure 1.

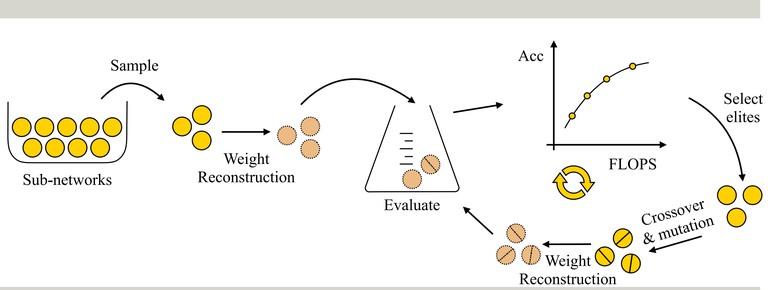

Figure 1. EAPruning Pipeline

Our pruning space for an attention block in Vision Transformers is shown in Figure 2.

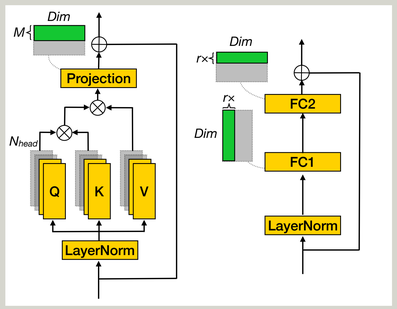

Figure 2. We prune the number of heads for Q, K and V (correspondingly the projection dimension) and MLP inner dimension only. The dotted gray area shows weights to be pruned, green area shows the remaining weights for convolutions. The shape of input and output of each block is retained.

Algorithm

Algorithm 1 Evolutionary Pruning Algorithm

Input: Original Network: $\mathcal{N}$, Pretrained Weights: $\mathcal{W}$, Population Size: $\mathcal{P}$, Number of Mutation: $\mathcal{M}$, Number of Crossover: $\mathcal{S}$, Max Number of Iterations: $\mathcal{T}$.
Output: $\mathcal{K}$ optimal Sub-Networks: $\mathcal{G}_{\mathcal{K}}$.

$\mathcal{G}_0$ = Random($\mathcal{N}$, $\mathcal{P}$);
for $i = 1 : \mathcal{T}$ **do**
  $\mathcal{G}_{metric}$ = Infer(Reconstruct($\mathcal{G}_{i-1}$, $\mathcal{W}$));
  $\mathcal{G}_i$ = NSGA-III.NextGen($\mathcal{G}_{metric}$, $\mathcal{M}$, $\mathcal{S}$);
end for
$\mathcal{G}_{\mathcal{K}}$ = ParetoFront($\mathcal{G}_{\mathcal{T}}$, $\mathcal{K}$);
return $\mathcal{G}_{\mathcal{K}}$;

Experimental Results

We prune on DeiT-Base, ResNet50 and MobileNetV1 with EAPruning and report their results in Table 1, 2 and 3. Improved speedup is shown in Table 4.

Model	FLOPs	Reduction	Top-1	Epochs	Training
DeiT-Base ([9])	17.8G	-	81.8%	300	Scratch
VTP ([11])	13.8G	22.4%	81.3%	100	Finetune
EAPruning (Ours)	13.5G	24.2%	81.3%	100	Finetune
AutoFormer ([1])	11.0G	38.2%	82.4%	500	Supernet
EAPruning (Ours)	11.0G	38.2%	81.6%	500	Scratch

Table 1. Pruning DeiT-Base on ImageNet, compared with state-of-the-art search-based methods.

Model	FLOPs	Reduction	Top-1
ResNet50 [4]	4111M	-	76.0%
AMC [5]	2047M	50.3%	75.5%
NetAdapt [10]	2239M	45.6%	75.9%
MetaPruning [7]	2G	51.4%	75.4%
N2NSkip [8]	≈2G	50%	74.6%
ResRep [3]	≈1871M*	54.5%	76.2%
EAPruning (Ours)	2019M	50.9%	75.7%
OTO [2]	≈1418M*	65.5%	74.7%
EAPruning (Ours)	1554M	62.2%	74.8%
EAPruning (Ours)	1063M	74.1%	73.6%

Table 2. Pruned ResNet50 on ImageNet at 2G and 1G FLOPs level. *: estimated by reduction ratio.

Model	FLOPs	Reduction	Top-1
MobileNetV1 ([6])	569M		70.6%
MobileNetV1 ([6])	325M	0.75×	68.4%
AMC ([5])	301M	0.89×	70.4%
NetAdapt ([10])	284M	1×	69.1%
MetaPruning ([7])	281M	1×	70.6%
MetaPruning ([7])	324M	0.75×	70.9%
EAPruning (Ours)	302M	0.88×	**71.1%**

Table 3. Comparison of pruned MobileNetV1 models on ImageNet.

Model	Throughputs (img/s)	Acc (%)	Speedup
ResNet50	3753		
ResNet50×0.5	5147	-0.30	**1.37**×
MobileNetV1	9176		
MobileNetV1×0.5	12296	-0.09	**1.34**×
DeiT-Base	777		
DeiT-Base×0.6	1086	-0.35	**1.40**×

Table 4. Our EA pruned models enjoys obvious speedup on NVIDIA A30 GPUs.

References

[1] Minghao Chen, Houwen Peng, Jianlong Fu, and Haibin Ling.
Autoformer: Searching transformers for visual recognition.
In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12270–12280, 2021.

[2] Tianyi Chen, Bo Ji, Tianyu Ding, Biyi Fang, Guanyi Wang, Zhihui Zhu, Luming Liang, Yixin Shi, Sheng Yi, and Xiao Tu.
Only train once: A one-shot neural network training and pruning framework.
In *NeurIPS*, volume 34, 2021.

[3] Xiaohan Ding, Tianxiang Hao, Jianchao Tan, Ji Liu, Jungong Han, Yuchen Guo, and Guiguang Ding.
Resrep: Lossless cnn pruning via decoupling remembering and forgetting.
In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4510–4520, 2021.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun.
Deep residual learning for image recognition.
In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[5] Yihui He, Ji Lin, Zhijian Liu, Hanrui Wang, Li-Jia Li, and Song Han.
Amc: Automl for model compression and acceleration on mobile devices.
In *Proceedings of the European conference on computer vision (ECCV)*, pages 784–800, 2018.

[6] 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.
arXiv preprint arXiv:1704.04861, 2017.

[7] Zechun Liu, Haoyuan Mu, Xiangyu Zhang, Zichao Guo, Xin Yang, Kwang-Ting Cheng, and Jian Sun.
Metapruning: Meta learning for automatic neural network channel pruning.
In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3296–3305, 2019.

[8] Arvind Subramaniam and Avinash Sharma.
N2nskip: Learning highly sparse networks using neuron-to-neuron skip connections.
In *BMVC*, 2022.

[9] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou.
Training data-efficient image transformers & distillation through attention.
In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021.

[10] Tien-Ju Yang, Andrew Howard, Bo Chen, Xiao Zhang, Alec Go, Mark Sandler, Vivienne Sze, and Hartwig Adam.
Netadapt: Platform-aware neural network adaptation for mobile applications.
In *Proceedings of the ECCV (ECCV)*, pages 285–300, 2018.

[11] Mingjian Zhu, Kai Han, Yehui Tang, and Yunhe Wang.
Visual transformer pruning.
arXiv preprint arXiv:2104.08500, 2021.