Network Compression via Cooperative Architecture Search and Distillation

Fanghui Xue
Department of Mathematics
University of California, Irvine
Irvine, CA, USA.
fanghuix@uci.edu

Jack Xin
Department of Mathematics
University of California, Irvine
Irvine, CA, USA.
jack.xin@uci.edu

***Abstract*—Neural Architecture Search (NAS) and its variants are competitive in many computer vision tasks lately. In this paper, we develop a Cooperative Architecture Search and Distillation (CASD) method for network compression. Compared with prior art, our method achieves better performance in ResNet-164 pruning on CIFAR-10 and CIFAR-100 image classifications, promising to be extended to other tasks.**

***Index Terms*—NAS, distillation, network compression, pruning**

I. Introduction

While neural networks have been widely used in a great number of academic and industrial scenarios, the huge computational costs for training and inference are not always affordable. Researchers have developed numerous network compression methods in order to increase efficiency. Many recent papers formulate this problem in a Neural Architecture Search (NAS) [1] framework. Network Slimming [2] has set up a group of scaling factors, which can indicate the significance of corresponding channels of features in the batch normalization layers of a convolutional neural network. This scaling tensor tends to be sparse when ℓ_1 regularization is applied. TAS [3] searches for the number of channels in each convolutional layer and the number of layers as well, using knowledge distillation to further increase network accuracy.

Some fast NAS algorithms can be utilized to search the optimal compressed networks. DARTS [4] splits the training dataset into two, and each half of the dataset is used to learn the weights and the architecture respectively, via a two-step gradient descent. Based on DARTS, TAS has further proposed a penalized loss to limit computational costs and manage the FLOPs. Although TAS is able to compress the network by around 30%, one shall be aware that *TAS performs much worse in accuracy than the un-pruned baseline for the task of CIFAR classification*. In addition, the 1st-order DARTS used in TAS has poor convergence while the 2nd-order alternative is slow and used less often. DARTS also suffers from the model collapse problem [5]. Motivated by these thoughts, the aim of this paper is to develop a Cooperative Architecture Search and Distillation *(CASD) method to improve accuracy appreciably for the compressed networks in a consistent and reliable manner*.

II. Methodology

In this section, we discuss our formulation of the network compression problem and the search, pruning and distillation algorithms.

When fitting a dataset with a neural network, typically we need to specify a model with several layers, set up a loss function, and learn the weight tensor with SGD or its variants. As our purpose now is to prune the network, we build the following framework accordingly:

1) Introduce a channel scoring tensor S and include it in the loss as a learnable parameter.
2) Learn the channel tensor S as well as the weight tensor w by some NAS algorithms.
3) Remove those channels with low scores (pruning).
4) Fine-tune the pruned network with knowledge distillation.

An easy way to impose the scoring tensor is to multiply the output of each channel C_{ij} by a significance indicator S_{ij} directly, where i and j stand for the indices of the layer and the channel. We adopt the method of Network Slimming [2], which delicately uses the scale of each channel in batch normalization layers as the score. With this learnable scoring tensor added, the loss becomes a function $\mathcal{L}(w, S)$, depending on both w and S. Our goal of Step 2) is find the optimal scoring tensor S so that the pruned network is optimal. This is clearly a special case of NAS, which can be solved by a group of NAS algorithms. In view of the convergence and efficiency problems of DARTS, we adopt a relaxed differentiable architecture search (RARTS [6]) method which proposed a three-step first order gradient descent to fix model collapse and speed up convergence. The relaxed Lagrangian to be minimized for model training is:

$$\mathcal{L}(v, w, S) = \mathcal{L}_1(w, S) + \lambda \mathcal{L}_2(v, S) + \frac{1}{2}\beta \|v - w\|_2^2,$$

where v is the tensor generated from the relaxation, sharing the same shape with w. $\mathcal{L}_1$ and $\mathcal{L}_2$ are the loss values computed on the two half splits of the training dataset. λ and β are hyperparameters to adjust the scale of the penalty terms. It has been verified in [6] that RARTS has achieved better accuracy than DARTS in various image classification tasks, and arrested the model collapse problem in training.

978-1-6654-3410-2/21/$31.00 ©2021 IEEE
DOI 10.1109/AI4I51902.2021.00018

In Step 3), we prune the channels whose scores are low, and obtain a compact network. Finally, we fine-tune this model with knowledge distillation [7]. Let $\hat{y}$ and y to be the logits of the unpruned baseline and the pruned network. We follow the settings of TAS [3], which basically penalizes the cross entropy loss with the KL divergence of the pruned network from the soft labels of the unpruned network:

$$\mathcal{L}_{fine} = \gamma \mathcal{L}_{CE} + (1-\gamma)\mathcal{L}_{KL},$$

where $L_{CE} = -\log \sigma_j(y)$ is the cross entropy loss when the target label is j, and

$$\mathcal{L}_{KL} = \sum_i \sigma_i(\hat{y}/T) \log\big(\sigma_i(\hat{y}/T)/\sigma_i(y/T)\big)$$

is the KL divergence. Here i goes over all the classification classes, and $\sigma_i(y) = \frac{\exp y_i}{\sum_k \exp y_k}$ is the softmax function. The hyper-parameters T and γ and determine the scale of the loss and the logits. Since the network parameters v and w are optimized in turn and promote each other via the ℓ_2 penalty (see Algorithm 1), it is called a cooperative method.

Algorithm 1: Cooperative Architecture Search and Distillation (CASD) for network compression

Input the learning rates η_w, η_v, η_S.
Initialize the weight tensors w^0, v^0 independently.
Initialize the channel scoring tensor S^0.
Optimize the loss $\mathcal{L}(w, v, S)$ by RARTS [6]:
while *not converged* **do**
  $v^{t+1} \leftarrow v^t - \eta_v^t \nabla_v \mathcal{L}(v^t, w^t, S^t)$
  $w^{t+1} \leftarrow w^t - \eta_w^t \nabla_w \mathcal{L}(v^{t+1}, w^t, S^t)$
  $S^{t+1} \leftarrow S^t - \eta_S^t \nabla_S \mathcal{L}(v^{t+1}, w^{t+1}, S^t)$
end
Prune the channels whose scores are low.
Fine-tune the network by distillation.

III. Experiments

We apply CASD to ResNet-164 [8] pruning on CIFAR-10 and CIFAR-100 [9] classification tasks. The hyperparameters of search and pruning follow those of Network Slimming: batch size = 256, learning rate = 0.1, weight day = 10^{-4}, and epoch = 160. The Pruning Ratios (PR) for Network Slimming and CASD are predetermined. That means, we rank the channel scoring parameters in each layer and prune the lowest 40% or 60% of them. The fine-tuning step is also similar, except for an extra KL divergence term for distillation. The hyper-parameters in the final distillation step follow those of TAS: the weight of cross-entropy loss vs. the weight of KL divergence = 9:1, the scaling temperature of the logits $T = 4$.

The baseline model in Table I is the unpruned ResNet-164. It has obtained the FLOPs of 2.48×10^8 with an error 4.22% on CIFAR-10, and the same FLOPs with an error 21.83% on CIFAR-100. In order to be consistent, the measure of FLOPs follows TAS, which is one half of the measure in Network Slimming. It is fair to assign TAS to the 40% PR group as its FLOPs is similar to the other two. Clearly, CASD beats Network Slimming and TAS in accuracy for both datasets.

TABLE I
Application of CASD to ResNet-164 pruning on CIFAR-10 and CIFAR-100, in comparison with the baseline (ResNet-164), TAS and Network Slimming (NS).

Data	Method	Test Error (%)	FLOPs
CIFAR-10	Baseline	4.22	2.48×10^8
	TAS [3]	6.00	1.78×10^8
	NS (40% PR) [2]	5.08	1.90×10^8
	CASD (40% PR)	**4.58**	1.90×10^8
	NS (60% PR) [2]	5.27	1.38×10^8
	CASD (60% PR)	**5.01**	1.33×10^8
CIFAR-100	Baseline	21.83	2.48×10^8
	TAS [3]	22.24	1.71×10^8
	NS (40% PR) [2]	22.87	1.67×10^8
	CASD (40% PR)	**21.63**	1.78×10^8
	NS (60% PR) [2]	23.91	1.24×10^8
	CASD (60% PR)	**22.38**	1.23×10^8

IV. Conclusion

We have developed a Cooperative Architecture Search and Distillation method for network compression. Thanks to the better search algorithm, it has achieved a comparable efficiency and better accuracy than prior state-of-the-art network compression methods for the ResNet-164 model on CIFAR. On CIFAR-100 in particular, CASD realized both 40% compression and better accuracy than the baseline ResNet-164. CASD can be easily generalized to any other datasets and networks with no changes other than data processing and model initialization. In future work, we also plan to further enhance knowledge distillation aspect of CASD.

Acknowledgments

The work was partially supported by NSF grant DMS-1854434, DMS-1952644, and a Qualcomm Faculty Award.

References

[1] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," *ICLR, 2017; arXiv preprint arXiv:1611.01578*, 2016.

[2] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang, "Learning efficient convolutional networks through network slimming," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2736–2744.

[3] X. Dong and Y. Yang, "Network pruning via transformable architecture search," in *Advances in Neural Information Processing Systems*, 2019, pp. 760–771.

[4] H. Liu, K. Simonyan, and Y. Yang, "DARTS: Differentiable architecture search," *ICLR, 2019; arXiv preprint arXiv:1806.09055*, 2018.

[5] X. Chu, T. Zhou, B. Zhang, and J. Li, "Fair darts: Eliminating unfair advantages in differentiable architecture search," in *European conference on computer vision*. Springer, 2020, pp. 465–480.

[6] F. Xue, Y. Qi, and J. Xin, "Rarts: a relaxed architecture search method," *arXiv preprint arXiv:2008.03901*, 2020.

[7] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *The Conference on Neural Information Processing Systems Workshop (NeurIPS-W)*, 2014.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[9] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," Citeseer, Tech. Rep., 2009.