Dynamic Slimmable Network

Supplementary Material

Changlin Li[1] Guangrun Wang[2] Bing Wang[3] Xiaodan Liang[4] Zhihui Li[5] Xiaojun Chang[1*]

[1] GORSE Lab, Dept. of DSAI, Monash University [2] Univeristy of Oxford [3] Alibaba Group

[4] Sun Yat-Sen University [5] Shandong Artificial Intelligence, Qilu University of Technology

changlin.li@monash.edu, wanggrun@gmail.com, fengquan.wb@alibaba-inc.com, xdliang328@gmail.com, zhihuilics@gmail.com, xiaojun.chang@monash.edu

Appendix

A. Implementation Details

Losses in Stage II. Complexity penalty loss $\mathcal{L}_{cplx}$ is used to increase the model efficiency in training stage II. To provide a stable and fair constraint, we use the number of multiply-adds on the fly, $\texttt{MAdds}(\mathcal{X},\theta)$, as the metrics of model complexity. Specifically, the complexity penalty is given by:

$$\mathcal{L}_{cplx}(\mathcal{X},\theta)=(\frac{\texttt{MAdds}(\mathcal{X},\theta)}{\mathbf{T}})^2, \tag{1}$$

where $\mathbf{T}$ is a normalize factor set to the total MAdds of the supernet in our implementation. Note that this loss term always pushes the gate to route towards a faster architecture, towards an architecture with target MAdds, which can effectively prevent routing easy and hard instances to the same architecture.

Overall, the slimming gate can be optimized with a joint loss function:

$$\mathcal{L}(\mathcal{X},\theta)=\lambda_1\mathcal{L}_{cls}+\lambda_2\mathcal{L}_{cplx}+\lambda_3\mathcal{L}_{SGS}. \tag{2}$$

The three balancing factors are set to $\lambda_1=1, \lambda_2=0.5, \lambda_3=1$ in our experiments. Different target MAdds is reached by adjusting the routing space during gate training. For instance, when training the gate of DS-MBNet-S, we set $\rho\in[0.35:0.05:0.5]$ to prevent routing to heavier sub-networks.

Equispaced channel group. Following previous works [14, 13], we set the the smallest division of channel number to 8. When using 0.05 as the interval of ρ, rounding channels by 8 may result in different intervals, which could lead to training failure when using Group Normalization [11]. To prevent such problem, we always adopt a consistent interval (*e.g.* 8, 16, 32) in a single layer, instead of multiplying ρ and rounding the channel. This results in a difference of the slimming ratio between our implemented architecture and our design.

*Corresponding author.

Additional details. Weight decay is set to 1^{-4} in all of our experiments on ImageNet. To stablize the optimization, weight decay of all the layers in the dynamic gate is removed. The weight γ of the last normalization layer of each residual block is initialized to zeros following [15]. The weight of the fully-connected layer in channel attention head, $\mathbf{W}_3$ in Eqn. 9 of the main text, is also zero-initialized to ease the optimization following [12]. Additional training techniques include [2, 1]. We do **not** use label smoothing [7], DropPath [6] and RMSProp [10], which are popularly used in previous works [9, 4, 13, 14].

B. Experiments on EfficientNet

We also applied our method on EfficientNet [9], a state-of-the-art network family with high efficiency. Similar to our DS-MBNet, **D**ynamic **S**limmable **Eff**icient**Net**-B0 (**DS-EffNet**) has only one slimming gate after its 8-th inverted residual block, controlling the rest 8 blocks. The fixed slimming ratio for the first 8 blocks is 0.5, while a uniform dynamic slimming ratio $\rho\in[0.75:0.05:1.75]$ is used for the last 8 blocks. This supernet with 20 paths in total is trained with a similar config with the supernet of DS-ResNet and DS-MBNet.

We train the supernet with 512 total batch size using 0.2 learning rate that decays with a cosine scheduler in 150 epochs. To enable direct comparision, we opt to reproduce the EfficientNet results using our training setup, with a 150 epoch schedule and no extra enhancement of DropPath [6], RMSProp [10], *etc.*

The result is shown in Tab. 1. DS-EffNet outperforms the original EfficientNet-B0 by 0.7% and 0.8%, proving its efficacy on recent methods with inverted bottleneck blocks [8] and Squeeze-and-Excitation module [5].

C. Additional Ablations

Slimming gate. We analysis the improvement brought by slimming gate by comparing the performance of DS-Net and its supernet. As shown in Tab. 2, slimming gate boosts the

Table 1. Comparison of EfficientNet-B0 and DS-EffNet on ImageNet.

	Method	MAdds	Top-1 Acc.
400M MAdds	EffNet-B0 [9] (repro.)	399M	76.0
	DS-EffNet-L (Ours)	400M	**76.7**
200M MAdds	EffNet-B0 0.75× [9]	267M	74.6
	DS-EffNet-S (Ours)	270M	**75.4**

Table 2. Ablation analysis of slimming gate.

model	MAdds	Top-1 Acc.
supernet (DS-MBNet)	140M	69.3
DS-MBNet-S	153M	70.1
supernet (DS-ResNet)	1.1B	73.4
DS-ResNet-S	1.2B	74.6

Table 3. Ablation analysis of distillation temperature τ (40 epochs).

τ	slimmest	widest
1	59.2	65.6
4	49.0	67.6

performance of DS-MBNet-S and DS-ResNet-S by 0.8% and 1.2% respectively, comparing to sub-networks with similar sizes in their supernet.

Distillation temperature. Temperature τ in distillation loss was first introduced in [3] to control the smoothness of the target. Using a properly larger τ usually yields better performance of the student. Surprisingly, we find a huge performance degradation in the slimmest sub-network when using larger τ in in-place distillation. We test $\tau = 4$ with DS-MBNet for 40 epochs and compare the it with the performance of default setting ($\tau = 1$). As shown in Tab. 3, the performance of the slimmest sub-network decrease by 10.2% after applying the temperature $\tau = 4$.

References

[1] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *CVPR*, 2019. 1

[2] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv:1706.02677*, 2017. 1

[3] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *arXiv:1503.02531*, 2015. 2

[4] A. Howard, M. Sandler, Grace Chu, Liang-Chieh Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, Vijay Vasudevan, Quoc V. Le, and H. Adam. Searching for mobilenetv3. In *ICCV*, 2019. 1

[5] J Hu, L Shen, S Albanie, G Sun, and E Wu. Squeeze-and-excitation networks. *IEEE transactions on pattern analysis and machine intelligence*, 2019. 1

[6] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Fractalnet: Ultra-deep neural networks without residuals. In *ICLR*, 2017. 1

[7] Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. Regularizing neural networks by penalizing confident output distributions. In *ICLR*, 2017. 1

[8] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018. 1

[9] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019. 1, 2

[10] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012. 1

[11] Yuxin Wu and Kaiming He. Group normalization. In *ECCV*, 2018. 1

[12] Zongxin Yang, Linchao Zhu, Yu Wu, and Yi Yang. Gated channel transformation for visual recognition. In *CVPR*, 2020. 1

[13] Jiahui Yu and Thomas Huang. Autoslim: Towards one-shot architecture search for channel numbers. *arXiv:1903.11728*, 2019. 1

[14] Jiahui Yu and Thomas S. Huang. Universally slimmable networks and improved training techniques. In *ICCV*, 2019. 1

[15] Jiahui Yu, Pengchong Jin, Hanxiao Liu, Gabriel Bender, Pieter-Jan Kindermans, Mingxing Tan, T. Huang, Xiaodan Song, and Quoc V. Le. Bignas: Scaling up neural architecture search with big single-stage models. In *ECCV*, 2020. 1