Data-Free Neural Architecture Search via Recursive Label Calibration - Appendix

A Details of Consistency Exploration

For DARTS search space, we randomly sample 100 architectures, and train each architecture from scratch using original CIFAR-10 data, synthetic data, or random noise separately and report the architecture's accuracy when evaluated on original CIFAR-10 validation dataset. When training on original CIFAR-10, we use cross-entropy loss. For the synthesized data and random noise, we use the KL-divergence loss between the network's output and the soft labels.

We use the SPOS [3] search space to explore the more challenging task of architecture search on the large-scale ImageNet dataset. We randomly sample 1000 architectures from the SPOS search space, then evaluate their accuracy. Since training thousands of architectures from scratch on the ImageNet dataset is computationally prohibitive, we train a SuperNet, which is originally proposed in SPOS, on different data sources including random noise, synthesized data, and ImageNet. We then evaluate the accuracy of 1000 randomly-sampled architectures on each data source. Since the SuperNet contains all the sub-networks and SPOS search algorithm itself uses the accuracy inferred from the SuperNet to rank different architectures in the search stage, a good correlation between the accuracy inferred from the SuperNet trained on synthetic data and the SuperNet trained on ImageNet provides a solid foundation for whether SPOS can discover high-quality architectures using synthetic data.

B Training Details of Data-free NAS

Our experiments on DARTS are targeted at classification over CIFAR-10 [5] dataset and our experiments on SPOS and ProxylessNAS are targeted at classification over ImageNet [11] dataset.

B.1 Data synthesis

For synthesizing data targeting the CIFAR-10 dataset, we use ResNet-34 trained on the CIFAR-10 data as the pre-trained model. We generate 50k 40×40 images, with a batch-size of 250 images per generation. We generate the images for each batch by running our optimization procedure for 2000 iterations in inner loop and 10 iterations in outer loop. We use Adam optimizer with a learning rate of 0.1. When targeting the ImageNet dataset, we use ResNet-50 trained on ImageNet as the pre-trained model. Then we generate 140k 256×256 images, with a batch-size of 50 images per generation, we use Adam optimizer with learning rate 0.25 to optimize each batch for 5000 iteration in inner loop and 10 iterations in outer loop. As mentioned in [6], the primary goal of NAS is to identify a high-quality network architecture. Training the weights of that architecture is not necessarily the objective of a NAS algorithm, which instead, is done during a separate evaluation phase. Thus, to testify that the data-free NAS can discover an architecture that performs well when trained and tested on the target data, and for a fair comparison with other state-of-the-art NAS algorithms, we use the original data for evaluating the searched architectures. We train the searched architecture from scratch with the same data and training schemes as the instantiation NAS methods originally used.

B.2 Data-free DARTS

Search Space The DARTS search space is defined as a cell-based search space, where the search process finds cells that can be applied to form a network of L cells. Each cell is a directed acyclic graph (DAG) of N nodes, where each node is a latent representation. The search algorithm searches for the operation of type o between two nodes from the operation space $\mathcal{O}$. The operation space $\mathcal{O}$ consists of: 3×3 and 5×5 separable convolutions, 3×3 and 5×5 dilated separable convolutions, 3×3 max pooling, 3×3 average pooling, identity, and *zero*. More details are described in the original DARTS paper [7].
Searching Phase We use synthetic CIFAR-10 data for data-free DARTS search. It contains 50k 40×40 images, *i.e.*, 5k images in each class. We use random crop augmentation. In each iteration, a 32×32 region is selected to be the input to the DARTS algorithm. The KL-divergence loss is imposed between the output logits of DARTS and the soft-label of the synthetic images for training data-free DARTS. A DARTS convolution cell consists of 7 nodes and a network of 8 cells. The initial number of channels is set to 16. DARTS alternatively updates the architecture parameters and weight parameters. In training data-free DARTS, we follow the training hyper-parameters in original DARTS paper [7] exactly. We hold out half of the synthetic training data as the validation set for search. For the weight parameters we use SGD with momentum 0.9, batch size 64, initial learning rate 0.025, and weight decay 3×10^{-4}. The weights are trained for 50 epochs. For architecture variables we use zero initialization and Adam with initial learning rate 3×10^{-4} and weight decay 10^{-3}.
Evaluation Phase In the evaluation phase, following [7], the network consists of 20 searched cells and 36 initial channels. It is trained for 600 epochs with batch size 96. Other hyper-parameters are the same as the hyper-parameters used in the search phase.

B.3 Data-free SPOS

Search Space The Single Path One-Shot (SPOS) [3] search space is a block-wise search space where each block can choose a different block o from the operation space $\mathcal{O}$. The operation space $\mathcal{O}$ contains 4 candidates: the ShuffleNet v2 [14] block with 3×3, 5×5, or 7×7 convolution, and the Xception block [2]. For more technical details, please refer to the original SPOS paper [3]

SuperNet Training Phase We use our synthetic ImageNet data for data-free SPOS. We randomly sample 32 synthetic images from each of the 1000 classes to form the validation dataset, which is used for validation during the evolutionary search. The SuperNet is trained using the remaining 108k synthetic images. We use the KL-divergence loss between SuperNet output logits and the soft-labels to train the SuperNet. The training scheme follows SPOS [3]. We train the SuperNet for 120 epochs using SGD with momentum 0.9, weight decay 4×10^{-5}, batch size 1024, and initial learning rate 0.5.

Searching and Evaluation Phase We conduct the evolutionary search with the accuracy of candidate architectures inferred on the synthetic validation set using the weights of the trained SuperNet. Then in evaluation, we train the best architecture from the search phase from scratch. We train for 240 epochs and all other training hyper-parameters are the same as used in training the SuperNet.

B.4 Data-free ProxylessNAS

Search Space The ProxylessNAS search space is based on a MobileNet V2 [12] backbone. It is a block-wise search space, searching for a mobile inverted bottleneck convolution (MBConv) in each block of the backbone. The candidate operations are chosen among MBConvs with various kernel sizes {3, 5, 7} and expansion ratios {3, 6}, as detailed in the original ProxylessNAS paper [1]

Searching and Evaluation Phase ProxylessNAS [1] trains an over-parameterized network that contains all candidate paths and uses architecture parameters to learn to select among the paths. When training the over-parameterized network, we form a validation set by randomly holding out ∼32k images from the synthetic training dataset. Instead of using the original loss, we impose the KL-divergence loss between the output logits of the over-parameterized network and the soft-labels of the synthetic images. We use the same hyper-parameter settings as the RL-based ProxylessNAS, which uses Adam optimizer with a 0.001 learning rate for training the architecture parameters. Then in evaluation, the searched network is trained from scratch with Adam optimizer for 300 epochs.

C Training Details of Extension Tasks

C.1 Data-free Pruning

Network channel pruning has been recognized as an effective network compression technique [4,9,10]. Traditional pruning requires original training data to perform pruning. In this study, we conduct pruning without the original training data by integrating our data-free NAS with a search-based pruning method named MetaPruning [8] and show the generalization ability of the proposed data-free NAS framework to the pruning tasks.

Data Preparation: We use synthesize 140k images generated from a ResNet-50 network pre-trained on the ImageNet dataset. We randomly split this synthetic data into two parts: 32k images for validation during search (*i.e.*, 32 images for each class) and the remaining 108k images are used as the training set.

Pruning Framework: The optimization framework for training PruningNet, evolutionary search, and evaluating the searched network follows MetaPruning. We first train a ResNet-50-based PruningNet from scratch on the synthetic training set. PruningNet is proposed in MetaPruning to generate weights for the pruned structures. We use the KL-divergence loss between the PruningNet output logits and the soft labels. Then we infer the accuracy of the candidate pruned networks from the synthetic validation set with the weights generated by PruningNet and use evolutionary search to find the best pruned network. After that, we train the best pruned network from scratch using the 140k synthetic images and the corresponding soft labels.

Hyper-parameters: We adopt the same training hyper-parameters as [8]. We randomly crop a 224×224 region from the 256×256 images and train the PruningNet for 32 epochs with an initial learning rate of 0.1 and batch-size of 256. We use SGD with a momentum of 0.9 and weight decay of 1×10^{-4}. In training the pruned network from scratch, we again use the synthetic images. Hyper-parameter settings remain the same as those used for training the PruningNet, except the number of training epochs is set to 128.

C.2 Data-free Knowledge Transfer

Further, we study knowledge transfer from teacher network to student network without using any original data. For a fair comparison with Dreaming to Distill [13], we use a pre-trained ResNet-50 network as the teacher network for image synthesis and train another randomly initialized ResNet-50 from scratch using 140k images synthesized with our proposed methods. We use the synthesized images and the teacher network's predictions on the images as soft-labels to train the student network and train using SGD with momentum 0.875, weight decay 3×10^{-5}, batch-size 1024 and initial learning rate 1.024.

References

1. Cai, H., Zhu, L., Han, S., et al.: Proxylessnas: Direct neural architecture search on target task and hardware. arXiv preprint arXiv:1812.00332 (2018) 3
2. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on CVPR. pp. 1251–1258 (2017) 3
3. Guo, Z., Zhang, X., Mu, H., Heng, W., Liu, Z., Wei, Y., Sun, J.: Single path one-shot neural architecture search with uniform sampling. arXiv preprint arXiv:1904.00420 (2019) 1, 2, 3
4. He, Y., Zhang, X., Sun, J.: Channel pruning for accelerating very deep neural networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1389–1397 (2017) 3
5. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Tech. rep., Citeseer (2009) 1
6. Liu, C., Dollár, P., He, K., Girshick, R., Yuille, A., Xie, S.: Are labels necessary for neural architecture search? arXiv preprint arXiv:2003.12056 (2020) 2

7. Liu, H., Simonyan, K., Yang, Y., et al.: Darts: Differentiable architecture search. In: International Conference on Learning Representations (2019) 2
8. Liu, Z., Mu, H., Zhang, X., Guo, Z., Yang, X., Cheng, K.T., Sun, J.: Metapruning: Meta learning for automatic neural network channel pruning. In: Proceedings of ICCV. pp. 3296–3305 (2019) 3, 4
9. Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., Zhang, C.: Learning efficient convolutional networks through network slimming. In: Proceedings of ICCV. pp. 2736–2744 (2017) 3
10. Liu, Z., Sun, M., Zhou, T., Huang, G., Darrell, T.: Rethinking the value of network pruning. arXiv preprint arXiv:1810.05270 (2018) 3
11. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision **115**(3), 211–252 (2015) 1
12. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4510–4520 (2018) 3
13. Yin, H., Molchanov, P., Alvarez, J.M., Li, Z., Mallya, A., Hoiem, D., Jha, N.K., Kautz, J.: Dreaming to distill: Data-free knowledge transfer via deepinversion. In: Proceedings of the CVPR. pp. 8715–8724 (2020) 4
14. Zhang, X., Zhou, X., Lin, M., Sun, J.: Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6848–6856 (2018) 2