AN EFFICIENT METHOD FOR MODEL PRUNING USING KNOWLEDGE DISTILLATION WITH FEW SAMPLES

ZhaoJing Zhou[⋆] *Yun Zhou*[†] *Zhuqing Jiang*[⋆] *Aidong Men*[⋆] *Haiying Wang*[⋆]

[⋆] School of Artificial Intelligence, Beijing University of Posts and Telecommunications
[†]Academy of Broadcasting Science, National Radio and Television Administration

ABSTRACT

Deep neural network compression methods can produce small-scale networks and utilizes fine-tuning to get back the dropped accuracy. Despite their remarkable performance, the fine-tuning procedure is limited to the requirement of a huge training dataset, which is a time-consuming progress. To address the issue, few-sample knowledge distillation (FSKD) has been proposed for data efficiency. However, FSKD needs to add additional convolution layers for compressed networks during training, which increases the complexity of network structure. In this paper, we present Progressive Feature Distribution Distillation (PFDD) without modifying network structures, which surpasses FSKD. Concretely, it is based on a progressive training strategy that is efficient for matching feature distributions between compressed network and original network. Thus, we can notably exploit both external information from samples and internal information from network, where using a small proportion of training dataset can yield quite considerable results. Experiments on various datasets and architectures demonstrate that our distillation approach is remarkably efficient and effective in improving compressed networks' performance while only few samples have been applied.

Index Terms— Network compression, knowledge distillation, few samples

1. INTRODUCTION

In recent years, convolution neural networks(CNN) have achieved the state of the art performance in a variety of computer vision tasks, ranging from image recognition to object detection[1, 2]. However, CNN models often bring in huge computation complexity and storage costs which is an enormous obstacle to apply such networks in some resource-limited scenarios directly.

To break this limitation, many researchers have devoted great interest to accelerate or compress neural networks[3, 4, 5]. Among them, network pruning draws a huge amount of attention due to its competitive performance and convenience. Pruning aims to obtain a small network by removing less essential weights in an outstanding but extensive network. Experiments show that the more weights have been pruned, the more accuracy drops rapidly[6]. Then fine-tuning is used to recover the pruned network's accuracy. However, current fine-tuning practices require the fully annotated training dataset and take a lot of training time, which would encounter many obstacles in actual implementation. For instance, medical data is inaccessible in most cases due to data privilege and privacy issues. Consequently, when we use the pruning method to optimize a large-scale model for practical deployment, the existing methods will be difficult to recover its lost accuracy where few training samples are available.

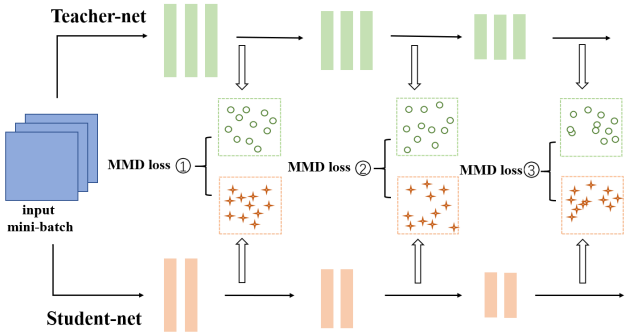

Fig. 1. It is the structure of our Feature Distribution Distillation. We show that teacher-net utilizes MMD loss to distill feature distribution into student-net at different positions. The pictures of dots and stars imply the activation maps of intermediate layers, representing the different distribution of the feature maps of the two networks.

To solve the recovery training problem in the few-sample scenario, FSKD[7] has been proposed for realizing knowledge distillation with few samples. However, FSKD has a few limitations. First, it adds extra convolution layers to fit the block-level outputs of the compressed network to the origin network, which increases the complexity of network. Also, it directly aligns the output of network's middle layer, which is

This work is sponsored by the National Key Research and Development Program under Grant (2018YFB0505200), National Natural Science Funding (No.62002026) and MoE-CMCC "Artificial Intelligence" Project under Grand MCM20190701.

easily disturbed by the redundant information in the original network. Eventually, it does not pay attention to mining the external information from samples and ignores the discrepancy between multiple instances[8].

This paper proposes a new distillation method to solve these issues, namely progressive feature distribution distillation (PFDD). We regard the original model as teacher-net and the compressed model as student-net. Inspired by the observation that the distribution of feature maps can reflect the discriminative image regions to help identify its category, we claim that feature distribution is valuable knowledge for promoting the performance of student-net. Based on this philosophy, we encourage student-net to mimic teacher-net's feature distribution from multiple instances, which can effectively measure the discrepancy between different samples. Therefore, we exploit the Maximum Mean Discrepancy (MMD)[9] metric as distillation loss to train the distribution of activations of student-net's intermediate layer to match that of the teacher. The illustration of our method for knowledge distillation is depicted in Fig. 1. Meanwhile, through using ingenious kernel function[10], MMD is convenient to apply without adjusting network structures. More importantly, to adapt to the few-sample scenario, we propose a progressive training strategy for the distillation. Instead of directly optimizing student-net based on all distillation losses, we manage to reduce the composition of distillation losses progressively. As a result, it provides a better opportunity to explore the internal relationship between teacher-net and student-net, which improves the training performance.

To summarize, our contributions are listed as follows:

• We present a novel distillation framework named progressive feature distribution distillation, which can recover a compressed network's performance with few samples.

• By matching the feature distribution from multiple instances, our method can leverage external information from samples and internal information from networks.

• Sufficient experiments show the effectiveness of our framework on various pruning methods and datasets. Also, it is convenient to be extended to other pruning methods.

2. PROPOSED METHOD

2.1. Feature Distribution Distillation

In this paper, we focus on exploring the internal representations. Fig. 2 shows some class activation maps(CAM)[11] from ResNet50 (trained on ImageNet, 75.99% top-1 val accuracy), on the left for middle layer and on the right for top pre-average pool. We can see that with the deepening of convolution layers, the feature distribution is gradually changing, paying more attention to those discriminative object areas, e.g., bird's head and wing, the outline of the Leaning Tower of Pisa and the cathedral. It is obvious that a class activation map for a particular category indicates the discriminative

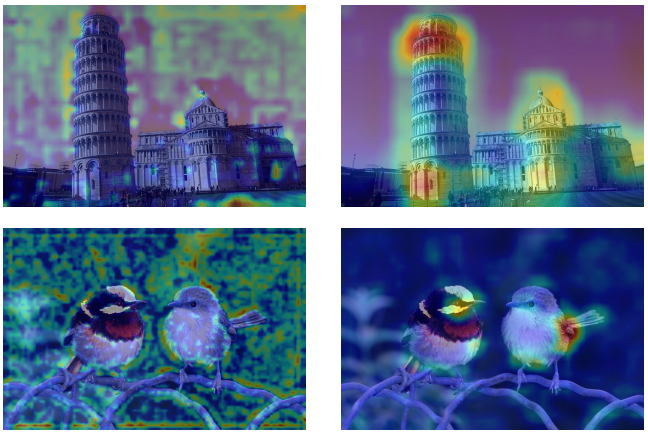

Fig. 2. Class activation maps for ResNet50 from hidden layers with different depth. Left column: the middle layer's output, right colum: top pre-average pool's output.

image regions used by CNN to identify that category. For a trained network, different inputs would give rise to activation distribution with different emphasis. In other words, it is beneficial to guide the student network towards a configuration that results in a distribution similar to teacher network. To mimic these patterns in the student, we use the Maximum Mean Discrepancy as a new metric of knowledge distillation and call it Feature Distribution Distillation(FDD).

Defined by the disparity between the mean embedding on two sets of samples, MMD can be seen as a distance metric for two distributions. Suppose $\mathcal{X}$ and $\mathcal{Y}$ are two sets of samples $\mathcal{X} = \left\{\boldsymbol{x}^i\right\}_{i=1}^N$, $\mathcal{Y} = \left\{\boldsymbol{y}^j\right\}_{j=1}^M$ from two different distributions p and q. Let $\mathbf{F} \in \mathbf{R}^{B\times C\times H\times W}$ be the intermedia layers' output feature maps. Then referring the teacher-net as T and the student-net as S, respectively. Through applying the kernel function $k(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle$ and replacing $(\mathbf{f}_T, \mathbf{f}_S)$ with (x, y), the squared MMD distance between p and q can be formulated as:

$$
\begin{aligned}
\mathrm{MMD}^2[X, Y] &= \left\| \frac{1}{N}\sum_{i=1}^{N}\phi\left(\mathbf{x}_i\right) - \frac{1}{M}\sum_{j=1}^{M}\phi\left(\mathbf{y}_j\right) \right\|^2 \\
&= \frac{1}{C_T^2}\sum_{i=1}^{C_T}\sum_{i'=1}^{C_T} k\left(\mathbf{f}_T^i, \mathbf{f}_T^{i'}\right) \\
&+ \frac{1}{C_S^2}\sum_{j=1}^{C_S}\sum_{j'=1}^{C_S} k\left(\mathbf{f}_S^j, \mathbf{f}_S^{j'}\right) \\
&- \frac{2}{C_T C_S}\sum_{i=1}^{C_T}\sum_{j=1}^{C_S} k\left(\mathbf{f}_T^i, \mathbf{f}_S^j\right)
\end{aligned}
\tag{1}
$$

For similarity, our method utilizes second-order poly kernel[10]: $k(\boldsymbol{x}, \boldsymbol{y}) = \left(\boldsymbol{x}^\top \boldsymbol{y}\right)^2$. Thus, the MMD between

two feature maps can be simplified as:

$$\mathrm{MMD}^2\left(\mathbf{F}_T, \mathbf{F}_S\right) = \left\|G_S - G_T\right\|_2^2 \tag{2}$$

where $\|\cdot\|_2$ is the L2-normalization and G is the Gram matrix, with each item as: $g_{ij} = \left(\mathbf{f}^i\right)^T \mathbf{f}^j$. The Gram matrix is the inner product between the vectorized feature map and manages to reflect these feature correlations. To measure the discrepancy between different samples, we can calculate the Gram matrix in the "batch" dimension. Considering the mini-batch B, the final feature distribution distillation loss is written as follows:

$$\mathcal{L}_{\mathrm{FDD}}\left(\mathbf{F}_T, \mathbf{F}_S\right) = \frac{1}{B^2} \sum_{(i,j)\in\mathcal{F}} \left\| G_T^{(i)} - G_S^{(j)} \right\|_2^2 \tag{3}$$

As we have seen, L2-normalization is used to match the two matrices. However, different from normalizing each feature vector, normalization after calculating the Gram matrix promotes the success of distillation.

Fig. 1 illustrates the complete architecture of our feature distribution distillation. There are two steps to our proposed method. In step 1, the student-net is collected by pruning a trained network using a practical network pruning mechanism. In step 2, the MMD losses are extracted at different depths and distill the feature distribution into student-net. Generally, we select three nodes evenly to calculate the MMD loss according to the depth of network.

2.2. Progressive Feature Distribution Distillation

Different layers across a trained network capture different feature concepts. Recent research shows that CNN extract semantical features that become more complex and abstract as we move from the shallow to the deep of the network. Instead of distilling all the internal layers simultaneously, we also consider distilling knowledge progressively matching internal representation in a maximum difference fashion. Sparked by [12, 13], we consider the following manner:

Progressive Feature Distribution Distillation(PFDD). We propose to follow a training order from large discrepancy to slight discrepancy in a progressive manner. First, we calculate the MMD loss produced by different depth layers to propagate backward. Then we introduce a sorting operation to pick out remaining losses according to their importance, which is based on the value of losses. A larger MMD loss means a more considerable discrepancy between feature maps of teacher-net and student-net. So student-net should pay more attention to that bigger one. For instance, in Fig. 1, suppose $loss_1$ is the maximum value and $loss_2$ is the minimum value, and then the PFDD loss will be given by $loss_1 + loss_2 + loss_3 \Rightarrow loss_1 + loss_3 \Rightarrow loss_1$.

Table 1. Performance comparison on different networks obtained by network slimming. The symbol $*$ means the model is trained on CIFAR-10 and 50 samples are used on the recovery phase, and ‡ means 100 samples from CIFAR-100. Full fine-tune utilizes full training data. Note that the ratio(like 70%) means the portion of removed channels compared to the whole channels in the network.

Pruned model	**Acc(%) Before / After**	**FLOPs/ #Param**	**Time** (mins)
VGG19$*$	93.76 / -	7.97 / 20.04	52
70%+PFDD(ours)	18.71 / **93.89**	3.91 / 2.24	3.5
70%+FSKD	18.71 / 93.41	3.91 / 2.24	2
70%+LwM[14]	18.71 / 93.35	3.91 / 2.24	6
70%+fine-tune	18.71 / 93.01	3.91 / 2.24	3
70%+full fine-tune	18.71 / 93.78	3.91 / 2.24	40
ResNet164$*$	95.05 / -	4.99 / 1.71	392
60%+PFDD(ours)	44.04 / **94.11**	2.75 / 1.13	5
60%+FSKD	44.04 / 93.25	2.75 / 1.13	2.5
60%+LwM	44.04 / 76.62	2.75 / 1.13	7
60%+FitNet	44.04 / 92.02	2.75 / 1.13	4
60%+fine-tune	44.04 / 71.07	2.75 / 1.13	4
60%+full fine-tune	44.04 / 94.71	2.75 / 1.13	360
DenseNet40$*$	93.82 / -	5.33 / 1.07	192
70%+PFDD(ours)	68.09 / **93.54**	2.79 / 0.39	4.5
70%+FSKD	68.09 / 92.45	2.79 / 0.39	2.5
70%+LwM	68.09 / 82.50	2.79 / 0.39	5
70%+FitNet	68.09 / 89.52	2.79 / 0.39	3.5
70%+fine-tune	68.09 / 78.86	2.79 / 0.39	3.5
70%+full fine-tune	68.09 / 93.80	2.79 / 0.39	178
VGG19‡	72.20 / -	7.97 / 20.08	64
50%+PFDD(ours)	3.04 / **71.50**	5.01 / 4.96	3.5
50%+FSKD	3.04 / 69.13	5.01 / 4.96	3
50%+LwM	3.04 / 57.28	5.01 / 4.96	4
50%+fine-tune	3.04 / 51.94	5.01 / 4.96	3
50%+full fine-tune	3.04 / 71.56	5.01 / 4.96	59
DenseNet40‡	74.37 / -	5.33 / 1.1	200
80%+PFDD(ours)	63.25 / **74.01**	3.26 / 0.6	4.5
80%+FSKD	63.25 / 73.05	3.26 / 0.6	3
80%+LwM	63.25 / 72.74	3.26 / 0.6	7
80%+FitNet	63.25 / 72.56	3.26 / 0.6	3.5
80%+fine-tune	63.25 / 68.19	3.26 / 0.6	3.5
80%+full fine-tune	63.25 / 74.54	3.26 / 0.6	189

3. EXPERIMENTS

In this section, we use different network structures to conduct sufficient experiments on classification datasets with various network structures, including VGG[15], ResNet[16] and DenseNet[17]. We implement the code with PyTorch and operate all experiments on one NVIDIA 2080TI GPU.

We obtain the student-net from a network pruning method named Network Slimming[18]. In order to simulate the scene with few samples, we randomly select hundreds of samples

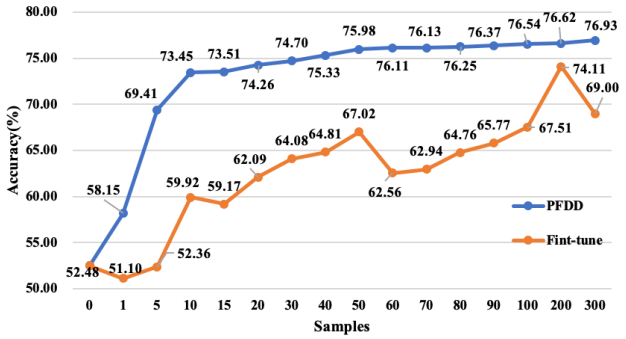

Fig. 3. It's the results of ResNet164(pruned 40%) trained with different numbers of samples on CIFAR-100. When the sample is 0, its accuracy(52.48%) represents the result of ResNet164 pruned 40% channels. And ResNet164's baseline accuracy is 77.27% trained on CIFAR-100.

(e.g., 50 or 100 samples) for the recovery phase from CIFAR-10 and CIFAR-100. This means each epoch only contains few samples.

We randomly combine the pruning ratio and network structures to experiment. Table 1 shows that no matter in which situations, our method has stronger adaptability and its results are far better than others. For example, for pruned 70% on VGG-19, PFDD exploits only 100 samples to recover the accuracy from 18.71% to 93.89%, which even outperforms the original model. It proves that PFDD can effectively excavate the discrepancy between multiple instances, which is better than aligning the output of network's middle layer directly (FSKD). In terms of the training time, it costs about 40 minutes for full fine-tuning to reach the equivalent level of accuracy as PFDD, while our method only takes few minutes to reach a similar level as full fine-tuning. Experiments conclude that PFDD is robust to different network situations and various pruning situations, and can realize both training efficiency(from the perspective of time) and sample efficiency.

Fig. 3 compares the performance of PFDD and fine-tune in different amounts of samples. On CIFAR-100, ResNet164 can achieve 77.27% validation accuracy. After pruning 40% redundant channels, it declines to 52.48%. It is illustrated that our method always outperforms fine-tune with a large margin and keeps a steady increasing trend. In extreme cases, PFDD still achieves outstanding results while fine-tune suffers extraordinary accuracy drops. Concretely, PFDD only needs 20 samples to recover the accuracy from 52.48% to 74.26%, which is hard for fine-tuning to reach. It proves PFDD combined with network pruning has a great deal of superiority on the limited data scenarios.

Owing to the student-net obtained by pruning the teacher-net, the weights of student-net are inherited from the teacher-net. More importantly, it does not require class label information of input data in the distillation training phase. That's why our method is able to quickly fit the distribution of the feature map to be consistent with the teacher-net, so as to efficiently recover the lost accuracy. Furthermore, the distillation mechanism can be seen as a type of regularization that helps prevent overfitting.

Table 2. Performance comparison between PFDD using samples from CIFAR-10 and CIFAR-100. FDD means distillation training doesn't adopt the progressive training manner.

	Samples	**Acc(%)** CIFAR-100	**Acc(%)** CIFAR-10
VGG19	50000	72.20	-
50%+PFDD	50	**71.17**	**69.32**
50%+FDD	50	70.46	66.61
50%+PFDD	100	**71.50**	**70.34**
50%+FDD	100	71.09	68.50

Since PFDD doesn't need label information, the following part discusses its availability when samples are some new images that the teacher-net has never seen during training. We evaluate PFDD's performance on VGG19 model trained on CIFAR-100 and pruned 50% channels, with the few samples are randomly selected from CIFAR-10 and CIFAR-100. Note that there is no intersection between the two datasets. As shown in Table 2, PFDD exceeds FDD by near 0.5% accuracy on CIFAR-100 and about 2% accuracy on CIFAR-10. The ablation study proves the significance of progressive training manner under a few-sample setting. At the same time, although PFDD doesn't require class label information of input images, it will slightly influence the performance using data from CIFAR-10. So that should make sense because familiar data can actively transfer valuable feature knowledge from teacher-net to student-net. However, PFDD is still able to achieve better performance than traditional fine-tuning even when the input images are totally new.

4. CONCLUSIONS

In this paper, a novel method of knowledge distillation named progressive feature distribution distillation is proposed, which is designed to alleviate the terrible recovery of the dropped accuracy when facing very few training examples. To be concrete, we exploit a progressive training strategy and use the MMD metric to measure the feature distribution of intermediate layers between student-net and teacher-net. The experimental results show that our method is not only better than the other knowledge distillation methods but also can achieve competitive results compared with full fine-tuning. In the future, we plan to excavate the potential of feature distribution distillation under the unsupervised scenario of typical image tasks.

5. REFERENCES

[1] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[2] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[3] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf, "Pruning filters for efficient convnets," *arXiv preprint arXiv:1608.08710*, 2016.

[4] 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.

[5] Qiulin Zhang, Zhuqing Jiang, Qishuo Lu, Jia'nan Han, Zhengxin Zeng, Shanghua Gao, and Aidong Men, "Split to be slim: An overlooked redundancy in vanilla convolution," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, 2020, pp. 3195–3201.

[6] Muyang Li, Ji Lin, Yaoyao Ding, Zhijian Liu, Jun-Yan Zhu, and Song Han, "Gan compression: Efficient architectures for interactive conditional gans," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5284–5294.

[7] Tianhong Li, Jianguo Li, Zhuang Liu, and Changshui Zhang, "Few sample knowledge distillation for efficient network compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14639–14647.

[8] Baoyun Peng, Xiao Jin, Jiaheng Liu, Dongsheng Li, Yichao Wu, Yu Liu, Shunfeng Zhou, and Zhaoning Zhang, "Correlation congruence for knowledge distillation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5007–5016.

[9] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan, "Learning transferable features with deep adaptation networks," in *International conference on machine learning*. PMLR, 2015, pp. 97–105.

[10] Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou, "Demystifying neural style transfer," *arXiv preprint arXiv:1701.01036*, 2017.

[11] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2921–2929.

[12] Gustavo Aguilar, Yuan Ling, Yu Zhang, Benjamin Yao, Xing Fan, and Chenlei Guo, "Knowledge distillation from internal representations.," in *AAAI*, 2020, pp. 7350–7357.

[13] Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han, "Once-for-all: Train one network and specialize it for efficient deployment," *arXiv preprint arXiv:1908.09791*, 2019.

[14] Prithviraj Dhar, Rajat Vikram Singh, Kuan-Chuan Peng, Ziyan Wu, and Rama Chellappa, "Learning without memorizing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5138–5146.

[15] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[16] 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*, 2016, pp. 770–778.

[17] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[18] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang, "Learning efficient convolutional networks through network slimming," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2736–2744.