Progressive Self-Knowledge Distillation with Mutual Learning

Sorn Chottananurak, Jongoh Jeong, Daeen Kabir and Thaweerath Phisannupawong*
Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea
{sorn111930, jeong2, dk2001, petchthwr}@kaist.ac.kr

Abstract

Deep neural networks based on large network architectures are often prone to the over-fitting problem and thus inadequate for generalization. Recent self-knowledge distillation (self-KD) approaches have successfully addressed this issue by regularizing a single network using dark knowledge (e.g., knowledge acquired from wrong predictions). Motivated by the idea of online collaborative learning using a large student cohort, we extend the online self-KD methods by combining the two learning schemes as in real-world learning environments. We seek to mimic the real-world self- and collaborative-learning strategies in deep neural networks for the image classification task, aimed to better predict the classification accuracy with lower computational costs during training. We closely explore the performance of a teacher-free dynamically evolving self-distilled network and verify that our approach on the CIFAR-100 dataset gives significant improvement from combining self-KD and mutual feature learning.

1. Introduction

In recent years, knowledge distillation (KD) has been successful in two main areas: neural network model compression and knowledge transfer from an over-parameterized teacher to the lightweight student model. However, the typical drawback of KD is that the student model mostly learns all knowledge from the teacher model; therefore, the student usually performs with little accuracy than the teacher. Moreover, the higher the teacher's performance, the more accurately the student can perform, so finding the best teacher is essential. Thus, ones may need to explore various architectures; therefore, the finding is usually computationally expensive. Instead of using a teacher model, self-distillation (self-KD) takes the output of more powerful parts as targets, then distills the knowledge to the shallower parts. However, it is still being determined whether the performance could be improved if the knowledge from the shallow layers can also be transferred to the deeper layer. Deep mutual learning (DML) is a successful method of two ways of transferring knowledge, enabling collaborative learning between students. This leads to the motivation that the neural layers can be treated as many students collaboratively learning instead of treating the deeper layer as a more potent teacher.

From both achievements of DML and self-KD, we explore the self-KD for improving the accuracy of a neural network without training the teacher model. Moreover, we examine DML when applied at the features level within the self-KD process. Our work demonstrates the comparative experiment conducted on the proposed method and previous works. Furthermore, we analyze the effect of DML in the self-KD pipeline applied at the feature level.

2. Related Work

2.1. Knowledge Distillation

Knowledge Distillation (KD) [3] has been used for neural network compression and knowledge transfer. In the teacher-student schema, the teacher is often powerful and over-parameterized, while the student is a relatively lightweight network. The KD is done by training the student on a transfer dataset, learning the soft target from the teacher to mimic the teacher's performance. As a result, the student performs significantly better in accuracy with its short inference time. Tian *et al*. [9] have presented the contrastive representation distillation (CRD), applying contrastive loss for knowledge transfer between networks. The objective pushes the representation of the same input closer and pushes away the representation from others; the student sometimes outperforms the teacher when applying CRD.

2.2. Self-Knowledge Distillation

Self-distillation transfers the knowledge within the model instead of transferring it across networks. Self-KD can be considered as a boosting technique for model training rather than model compressing or network knowledge transferring, for there are no high-performance teacher models involved. In [12], the network is split into sections dangled with classifiers. Then, the knowledge is transferred

*In alphabetical order of last names.

from the deeper to the shallower layers. The softmax targets that is set after the distilling sections are defined as,

$$\hat{p}_i(\mathbf{x};\tau) = \frac{\exp(z_i(\mathbf{x})/\tau)}{\sum_j \exp(z_j(\mathbf{x})/\tau)}, \quad (1)$$

where the temperature parameter, τ, scales the logit vector $z(\mathbf{x})$ in obtaining the predicted output probability $p(\mathbf{x})$. The self-distillation can also be implemented for self-supervised learning [1] by extracting the feature representation through a pre-trained model and performing pseudo-labeling via a clustering algorithm. The pseudo-labels were set as the soft-target to distill the knowledge to the separate network sections, similar to [12]. Kim *et al*. [4] demonstrate a Progressive self-knowledge distillation technique (PS-KD). Using PS-KD, the model is trained with the soft target, which is given by the adaptive combination of the label and the prediction in the previous epoch. As a result, the model trained with PS-KD achieved a high accuracy and high quality of confidence. Yun *et al*. [11] presented Class-wise Self-knowledge distillation that focuses on regularization by considering the distillation backward of the soft target in each class label. The work in [7] presented the 2-round self-KD that first trains the model from scratch, then distill the knowledge again to the current model along with ground-truth target twice. Apart from their method, the work has also shown that the students trained with self-KD sometimes outperform the teacher.

2.3. Teacher-free Online Collaborative Learning

Lan et al. [14] have demonstrated the online distillation method situated that the high-capacity teacher is not available. The process creates auxiliary branches in a network and trains all units simultaneously. That knowledge, the collective logits of all units, is transferred backward to individual branches online, constructing a strong generalizable teacher with lower training cost. Song et al. [8] presented collaborative learning for a network. Using this technique, multiple classifier heads are first added by keeping the original network structure. The classifiers are trained simultaneously using ground-truth labels. The gradient flows from these classifiers back through the whole network, resulting in a well-generalized, robust network without extra inference cost. Guo et al. [2] have presented the KD method via Collaborative Learning (KDCL). The technique considers all neural models as students, and all logits produced by each are combined into ensemble logits and then transferred back to all models. The method allows parallel computing to train faster and perform with high accuracy. Teacher-free knowledge distillation (Tf-KD) methods are proposed in [10]. The first method uses soft-target as regularization, and the second uses a virtual, manually designed teacher to teach a student. The technique achieved comparable results to regular vanilla KD.

3. Methodology

This paper seeks to leverage the dark knowledge of the *teacher* model in the self-knowledge distillation learning scheme. That is, we exploit a single network by dynamically evolving it to learn from its own softened target labels while collaboratively learning within the network across layer outputs.

3.1. Training Pipeline

We propose to incorporate the idea of cross-model collaboration of students within a single neural network through self-knowledge distilling, expecting more efficient and accurate training for image classification. For the architecture, a residual network (ResNet) is separated into many sections (ResBlocks) that collaboratively learn. Between each separated section, the classifiers are attached for knowledge distilling.

Implementing the PS-KD [4], the knowledge in the previous epoch is leveraged by considering the trained model in the previous epoch as a teacher. After that, the model prediction produced by the teacher is adaptively combined with the ground-truth labels making the distilling training labels for the network at present. These labels are distilled to each classifier of the student in the present epoch using the Progressive distillation loss $L_{CE} = CE\Big(P^S(\mathbf{x}), \mathbf{y}_{soft}\Big)$, /where $\mathbf{y}_{soft} = (1-\alpha)\mathbf{y} + \alpha P^T(\mathbf{x})$ with $\alpha = 0.8$. In the lower level, the self-KD [12] in the present epoch was implemented within the network. According to the self-KD framework, the present model also produces the prediction, and these predictions, combined with the hard labels, are the soft targets distilled to all dangled classifiers through the cross-entropy loss. In addition, the KL divergence is implemented utilizing the softmax output of the deepest classifier to ones from all shallower softmax output. Not only in the logits, but the L2 hint loss is also employed in the feature level of the classifiers, introducing the implicit knowledge of the deepest feature map to the shallower feature maps attached within the classifiers. These three loss sources are collected into $L_{Self-KD}$ defined as,

$$\begin{aligned} L_{Self-KD} = \sum_i^N \Big(& (1-\alpha) \cdot CE(P_i^S(\mathbf{x}), \mathbf{y}_{soft}) \\ & + \alpha \cdot D_{KL}(P_i^S(\mathbf{x}), P_N^S(\mathbf{x})) \\ & + \lambda \cdot ||\mathbf{F}_i - \mathbf{F}_N||_2^2 \Big), \end{aligned} \quad (2)$$

where N denotes the number of ResBlocks, CE is the cross-entropy loss, and D_{KL} is the KL-divergence. P is the predicted probability and $\mathbf{F}_i$ is the feature map at i-th layer. We set $\alpha = 0.1$ and $\lambda = 1 \times 10^{-6}$. Among all classifiers, we exploited minimizing the divergence of the mutual representation using deep mutual learning (DML) [13]. The

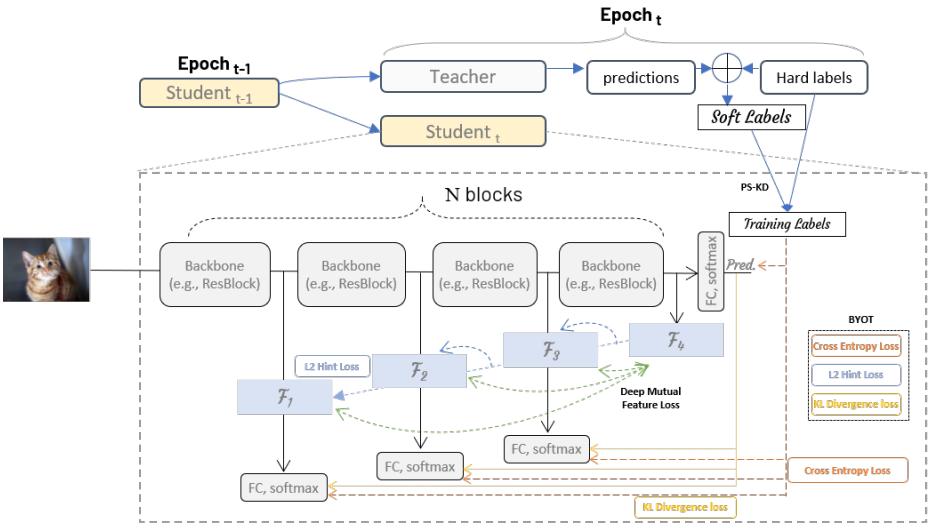

Figure 1. **Overview of the proposed learning pipeline.** We apply a set of different losses: Cross-entropy loss with softened labels from PS-KD (blue arrows on top), *self-distillating BYOT losses* (L2 hint loss in blue, KL divergence in yellow, and cross-entropy loss with soft targets from PS-KD in orange), and *deep mutual feature loss* (in green).

deep mutual feature loss derived from the KL divergence is employed between the deepest classifier's feature map and each classifier's feature map individually in both forward and backward directions, allowing the collaborative learning of the implicit knowledge within the network.

$$L_{DMFL} = \frac{1}{N-1}\sum_{i=1}^{N-1}\Big(D_{KL}(\mathbf{F}_i||\mathbf{F}_N)+D_{KL}(\mathbf{F}_N||\mathbf{F}_i)\Big) \tag{3}$$

The final learning objective L is then formulated as the sum of these losses as follows: $L = L_{CE} + L_{Self-KD} + L_{DMFL}$. The training pipeline is illustrated as a schematic diagram in Fig. 1.

4. Experimental Results

4.1. Implementation Details

Dataset: CIFAR-100 [5] contains 50K training images with 0.5K images per class and 10K test images. Both consist of 32 × 32 color images containing objects from 100 classes.

Network architectures: We conduct experiments on three different network architectures: ResNet-18, ResNet-50, and ResNeXt-50. We use a Nesterov SGD optimizer with learning rate of 0.1, momentum of 0.9, and a weight decay rate of 5×10^{-4}. We test with a batch size of 128 and apply random crop (32 with padding of size 4), random horizontal flip followed by ImageNet μ, σ normalization.

Evaluation metrics: We use the standard top-1 and top-5 errors to measure the performance for multi-class classification. In addition, we employ the negative log-likelihood (NLL), expected calibration error (ECE) [6], and the area under the risk-coverage curve (AURC) [1] to report the predictive probability in terms of confidence.

4.2. Results and Discussion

From the results in Table 1, intuitively, the training time of the large architecture is longer than small architecture. Considering the same architecture with multiple techniques, the training time increases when more techniques are implemented. This is because more gradients flowing requires more computational operation during the training. However, as the techniques facilitate the model optimization, sometimes combining the techniques trains the models faster. Comparing the implementations of the single self-KD technique, among three techniques, PS-KD performs better on the smaller model, in this case, ResNet-18, achieving significantly higher accuracy than BYOT and DMFL. However, in a larger model, the performance of those three methods is similar. When two self-KD methods are combined, the performance of these pairs slightly improves from the single-technique training. Combining three techniques has provided the most significant improvement from adding much more meaningful features into each classifier. Table 2 and Table 3 show the results of ablation studies on the number of self-KD blocks and the number of backbone blocks in the network. With fewer self-kd loss blocks, the top-1 and 5 accuracies tend to decrease. Moreover, the result shows 3-block network performs better in classification accuracy than the 2-block and 4-block networks.

Since we have a strict time limitation, all proposed ex-

Table 1. Experimental results on CIFAR-100 against other self-distillation methods. Network parameters: ResNet-18 (13.0 M), ResNet-50 (50.4 M), ResNeXt-50 (37.9 M). Note that the training time is the measured wall time until the best accuracy. Note that the training time is inconsistent due to heavy GPU load given a limited time. See per-epoch training times for fair comparison. The best result is **bolded**.

#	Model +Method	Top-1 Err (%)↓	Top-5 Err (%)↓	NLL	ECE (%)	AURC ($\times 10^3$)	Train Time (min./sec.)	Train time per epoch (sec.)
1	ResNet-18	21.41	5.57	0.87	5.14	56.92	944 / 05	25.3424
2	+BYOT	21.89	5.64	1.00	11.24	53.79	706 / 17	28.9306
3	+DMFL	21.61	5.45	0.87	5.31	57.13	872 / 25	25.6847
4	+PS-KD	**19.94**	4.79	0.79	4.21	50.44	622 / 05	27.0058
5	+PS-KD+BYOT	20.67	4.66	0.78	7.42	50.91	675 / 36	32.4176
6	+PS-KD+DMFL	20.02	4.73	0.80	4.52	50.55	585 / 23	27.0788
7	+DMFL+BYOT	21.18	5.47	1.02	11.74	54.89	718 / 08	31.7274
8	+PS-KD+BYOT+DMFL	20.03	**4.16**	**0.72**	**4.15**	**49.86**	559 / 11	32.0838
9	ResNet-50	21.96	5.21	0.89	8.75	57.24	1061 / 17	68.8616
10	+BYOT	19.03	4.38	0.94	11.51	48.32	2120 / 04	85.0151
11	+DMFL	20.13	4.96	0.85	8.63	51.20	1036 / 44	68.8262
12	+PS-KD	20.19	4.50	0.77	4.03	50.76	757 / 30	68.9272
13	+PS-KD+BYOT	17.92	3.94	0.75	8.96	46.76	2013 / 24	85.7228
14	+PS-KD+DMFL	19.12	4.43	0.73	**3.51**	47.57	2682 / 34	70.5304
15	+DMFL+BYOT	19.70	4.77	0.87	9.79	48.58	1369 / 31	85.8018
16	+PS-KD+BYOT+DMFL	**17.52**	**3.75**	**0.71**	10.00	**46.25**	965 / 51	86.4942
17	ResNeXt-50	19.25	4.48	0.81	7.49	48.13	3183 / 49	152.616
18	+BYOT	19.09	4.54	0.84	9.62	46.66	2907 / 59	164.871
19	+DMFL	19.41	4.43	0.82	7.45	47.72	1481 / 23	145.137
20	+PS-KD	18.79	4.20	0.74	**4.85**	48.52	3228 / 05	147.371
21	+PS-KD+BYOT	18.36	4.34	0.80	9.99	46.50	1396 / 17	165.050
22	+PS-KD+DMFL	19.25	4.14	0.75	5.18	47.46	1764 / 02	146.425
23	+DMFL+BYOT	19.30	4.23	0.88	9.86	47.98	1518 / 14	164.186
24	+PS-KD+BYOT+DMFL	**18.27**	**4.11**	**0.72**	10.32	**45.71**	1570 / 10	165.739

periments could not be done in time, and many experiments had to be conducted on different machines parallelly; therefore, the training time in this paper cannot be evaluated unbiasedly. Furthermore, the comparing models should achieve their highest capability on the given technique by close-to-optimal hyperparameters to claim that the method can outperform the existing works. Thus, carefully fine-tuning the hyperparameters is essential because they strongly influence the performance of the networks. Since the proposed training technique can be implemented in various network structures and datasets, experiments on both more architectures and datasets should be conducted. This is to present the more substantial confidence that the training method is efficient.

Table 2. Layer-wise top-1 accuracy (%) results on CIFAR-100.

Network Arch.	Classifier 4/4	Classifier 3/4	Classifier 2/4	Classifier 1/4	Ensemble
ResNet-18	78.59	0.92	1.24	0.60	78.08
+PS-KD+BYOT+DMFL	**79.76**	**67.90**	**75.22**	**78.92**	**79.68**
ResNet-50	79.04	1.12	0.81	0.89	78.78
+PS-KD+BYOT+DMFL	**81.98**	**72.64**	**79.11**	**81.80**	**82.14**
ResNeXt-50	80.74	1.25	1.06	0.84	80.53
+PS-KD+BYOT+DMFL	**81.59**	**78.22**	**79.76**	**81.67**	**82.98**

5. Conclusion

In this paper, we extend the online self-knowledge distillation methods by combining the two learning schemes as in real-world learning environments. We seek to mimic the real-world self- and collaborative-learning strategies in deep neural networks for the image classification task, aimed to better predict the classification accuracy with lower computational costs during training. We closely explore the performance of a teacher-free dynamically evolving self-distilled network and verify that the proposed techniques-combining approaches on the CIFAR-100 dataset outperform previous distillation methods.

For the recommendation for future works, because the training pipeline in this paper has some redundancies in the knowledge transferring, resulting in the additional gradient computation. Therefore, ones should consider analyzing the effects of KD loss functions and reformulating the transferring loss function accordingly. With a more relaxed time limitation, ones could consider experimenting on various architectures and datasets with well-fine-tuned hyperparameters. Moreover, this paper presented only the image classification task; therefore, future works could be implemented on other tasks, such as object detection and semantic scene labeling.

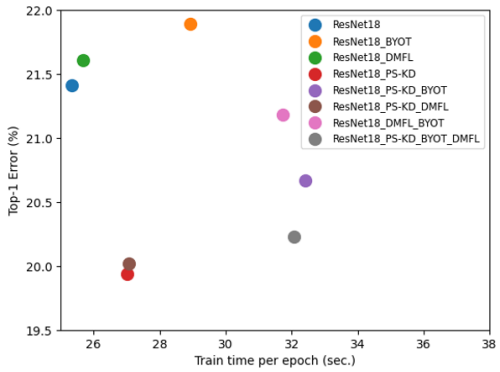

Figure 2. Top-1 error (%) vs. per-epoch training time (sec.)

Table 3. Ablation study on the BYOT features to which self-distilling BYOT losses are applied (ResNet-18). We report the validation accuracy for each ablated item. * indicates that there was some other load to the GPU affecting the training time.

#	PS-KD	DML	BYOT loss blocks	Top-1/5 err. (%)	Train time (min.)
1	✓	✓	1,2,3	20.23 / 4.56	559
2	✓	✓	2,3	20.57 / 4.93	1082*
3	✓	✓	3	21.21 / 5.20	625*

Table 4. Ablation study on ResNet-18 blocks with PS-KD+DMFL+BYOT. * indicates that there was some other load to the GPU affecting the training time.

#	N (# of backbone blocks)	Top-1/5 err. (%)	Train time (min. / sec.)
1	4	21.21 / 5.20	625 / 52
2	3	21.00 / 4.69	678* / 05
3	2	26.33 / 6.30	515 / 59

Acknowledgments

We appreciate the valuable advice and support by Professor Tae-Kyun Kim.

References

[1] Yonatan Geifman, Guy Uziel, and Ran El-Yaniv. Bias-reduced uncertainty estimation for deep neural classifiers. *arXiv preprint arXiv:1805.08206*, 2018. 2, 3

[2] Qiushan Guo, Xinjiang Wang, Yichao Wu, Zhipeng Yu, Ding Liang, Xiaolin Hu, and Ping Luo. Online knowledge distillation via collaborative learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11020–11029, 2020. 2

[3] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 1

[4] Kyungyul Kim, ByeongMoon Ji, Doyoung Yoon, and Sangheum Hwang. Self-knowledge distillation with progressive refinement of targets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6567–6576, 2021. 2

[5] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 3

[6] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29, 2015. 3

[7] Minh Pham, Minsu Cho, Ameya Joshi, and Chinmay Hegde. Revisiting self-distillation. *arXiv preprint arXiv:2206.08491*, 2022. 2

[8] Guocong Song and Wei Chai. Collaborative learning for deep neural networks. *Advances in neural information processing systems*, 31, 2018. 2

[9] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. *arXiv preprint arXiv:1910.10699*, 2019. 1

[10] Li Yuan, Francis EH Tay, Guilin Li, Tao Wang, and Jiashi Feng. Revisit knowledge distillation: a teacher-free framework. 2019. 2

[11] Sukmin Yun, Jongjin Park, Kimin Lee, and Jinwoo Shin. Regularizing class-wise predictions via self-knowledge distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13876–13885, 2020. 2

[12] Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3713–3722, 2019. 1, 2

[13] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4320–4328, 2018. 2

[14] Xiatian Zhu, Shaogang Gong, et al. Knowledge distillation by on-the-fly native ensemble. *Advances in neural information processing systems*, 31, 2018. 2