Online Knowledge Distillation with History-Aware Teachers

Chengcheng Li Zi Wang Hairong Qi
University of Tennessee, Knoxville, TN, USA
{cli42,zwang84}@vols.utk.edu, hqi@utk.edu

2022 International Joint Conference on Neural Networks (IJCNN) | 978-1-7281-8671-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/IJCNN55064.2022.9892064

***Abstract*—In this work, we propose a novel online knowledge distillation (OKD) approach, built upon the classical deep mutual learning framework in which peer networks (students) treat each other as teachers by learning from their predictions. The proposed method traces and leverages two levels of information encoded in each peer's learning trajectory to dynamically construct superior teachers to supervise other students. We first build a recurrent neural network associated with each peer, which takes both the network's current and previous logits as input and outputs integrated logits with the same dimension as the transferred knowledge. By doing so, the teachers provide an enhanced representation of knowledge. Beyond that, we also build a weight-averaged surrogate for each network, which maintains the exponential moving average of its learned parameters during the online training procedure. The proposed approach exploits the hidden information behind the online learning process instead of myopically learning from peers' outputs at a single time/iteration step. It potentially reduces uncertainties from peers as suffered in previous OKD studies with more stabilized transferred knowledge. We evaluate the proposed approach with benchmark image classification datasets and network architectures. Experimental results demonstrate its effectiveness with clear performance improvement over state-of-the-arts.**

***Index Terms*—online knowledge distillation, history-aware knowledge, network-based OKD, learning trajectory**

I. Introduction

Although deep neural networks (DNNs) have achieved state-of-the-art performance on many machine learning tasks, the issue of over-parameterization has prevented them from being deployed on resource-constrained edge devices [1], [2]. Network compression, naturally, becomes an essential research subject. Besides network quantization [3], pruning [4]–[7], and light-weight architecture design [8], knowledge distillation (KD) [9]–[11] is another popular branch of approaches for constructing a compact network structure without sacrificing accuracy. Classic KD trains a compact DNN (student network) by leveraging the transferred knowledge from a pre-trained, usually cumbersome DNN (teacher network). With the teacher's supervision, the student network usually exhibits superior performance than being trained independently.

In practice, the pre-trained powerful teacher does not always exist, making it difficult to utilize classic KD for model compression. To overcome this limitation, online knowledge distillation (OKD) [12]–[14] has been developed in recent years. OKD simultaneously trains a group of student networks from scratch. Each network considers its peers as teachers during training. Surprisingly, networks trained with OKD can achieve comparable, if not better, performance than their counterparts with the attendance of a powerful pre-trained teacher using classic KD.

Despite the huge success of OKD in training compact DNNs, there are still issues remain. Within the general OKD framework, a student learns its own parameters while serving as an instructor for its peers, introducing more uncertainty into the training procedure [15], [16]. Unpredictable behaviors, like oscillation, saturation, or failure, may happen [15], [17] in the middle of the training process. In this work, we argue that the history information collected during the course of learning/training can be utilized to yield a more stable outcome from OKD.

We propose to leverage the information encoded in the learning trajectory of peer networks during training to generate superior knowledge transferred through OKD. Specifically, we embody the learning trajectory at two levels, i.e., logits-level and weight-level, with two plug-and-play modules. In the first module, we employ a recurrent neural network (RNN) [18] associated with each peer network for the logits-level ensembling. In specific, the RNN takes a network's historical sequence of logits as well as current logits as the input and constructs integrated logits as a superior target for other peers to learn from. In the second module, we establish a surrogate model with the identical architecture for each student network, whose weights are replaced by the exponential moving average (EMA) of the student's weights at every iteration during training. Each student network takes supervision from the associated surrogates of peers instead of directly from the peers. The two proposed history-aware modules tend to distill knowledge [19]–[21] with more certainty, and as a consequence, potentially improve the performance of OKD.

Our main contribution is summarized as follows.

- Instead of learning with the knowledge generated at each single iteration step as used in existing approaches, we propose to leverage the history information to distill trajectory-aware knowledge for OKD.
- Two proposed modules, i.e. logits ensembling and weight ensembling, track the learning trajectory at two different levels, which enhance the quality of knowledge transferred and thus improve the performance.

This work is in part supported by National Science Foundation, CNS 2038922.

Authorized licensed use limited to: SUN YAT-SEN UNIVERSITY. Downloaded on November 27,2024 at 12:09:06 UTC from IEEE Xplore. Restrictions apply.
978-1-7281-8671-9/22/$31.00 ©2022 IEEE

- The proposed approach achieves state-of-the-art performance on the image classification task with multiple benchmark networks and datasets evaluated.

II. Related Work

A. Online Knowledge Distillation

Classic knowledge distillation (KD) approaches [22]–[26] aim to learn a compact student model using a pre-trained, over-parameterized teacher model as the guidance. Although KD effectively helps the student network to achieve higher performance, its success heavily relies on the existence of the pre-trained model. In contrast, online knowledge distillation (OKD) trains multiple peers from scratch, treating each other as the teacher while learning its parameters. OKD fills in the gap where a pre-trained teacher model is missing which is common in many practical scenarios while achieving surprisingly competitive performance.

Existing OKD approaches can be mainly categorized into two classes based on the learning structure [9], [27], i.e., branch-based approaches [13], [14] and network-based approaches [12], [28]–[30]. On the other hand, these methods (branch or network-based) can also be categorized based on the type of knowledge transferred [29], [31]–[34].

Branch-based OKD. Branch-based OKD approaches use a single network with multiple auxiliary branches that share lower-level layers. Each branch is considered as an individual model. Among this line of research, a representative work is ONE [14], which ensembles the auxiliary branches with a gating module to provide a stronger teacher. Following ONE, OKDDip [13] proposes a similar framework with two-level distillation, where an extra attention-based mechanism for the predictions of each branch is introduced to enhance peer diversity. Branch-based OKD approaches have achieved competitive performance, but they have some limitations. First of all, networks need to be manually redesigned to effectively ensemble information from branches and auxiliary branches need to be pruned after the training process. Second, it is difficult to ensemble student networks with significantly different architectures.

Network-based OKD. Network-based OKD approaches, on the other hand, are a more flexible scheme in which each network holds its own parameters. That is, students can have completely different structures. DML [28], as the first OKD mechanism, leverages the KL divergence loss between the logits of peers as the objective function in addition to the cross-entropy loss for the classification task. By doing so, the performance of networks outperforms those being trained independently with a clear margin. Chung et al. [29] proposed to use a discriminator associated with each model, which identifies the source of the feature maps and uses this information to transfer the feature representation with each other. Online collaborative learning is investigated in [12], which manipulates the training samples by adding noises to improve the peer networks' performance. [35] also presented a peer collaborative learning framework for OKD.

Some work presents both branch-based versions and network-based versions. For example, OKDDip [13] also has a network-based version with a similar strategy to diversify the outputs of peers. In this study, the proposed history-aware OKD is essentially a network-based framework.

B. Investigating Learning Process

Some literature has explored the potentials to use the information encoded in the learning process for specific tasks [36]. Laine and Aila [20] proposed to ensemble a model's outputs at different training stages to label the samples with unknown ground truths so that the model can be learned with only a small portion of labeled images in a semi-supervised manner. In [21], the authors discovered that by averaging weights of a model temporally instead of predictions, consistency target can improve the performance of semi-supervised learning. Flennerhag et al. [37] introduced a meta-learning framework, which aims to minimize the gradient paths in the learning procedure on task manifolds so that the learned model can be generalized to the new, unseen tasks with much fewer data.

Inspired by these studies, we propose to leverage the learning trajectory generated by each peer in the OKD process as an extra alignment among peers to enrich the knowledge learned by each student network. In specific, two modules are proposed that encode logits-level and weight-level information, respectively.

III. The Proposed Method

In this section, we elaborate on our proposed method that leverages the learning trajectory during the OKD training process to build history-aware "teachers". We first briefly present the overall framework and motivations. After the necessary preliminaries regarding KD and OKD, we describe the design of the two ensembling components in our approach.

A. Overview

Like other existing methods, [12], [28], the proposed OKD method is built upon the paradigm that the student networks are trained in a peer-teaching manner. Considering two student networks in the context of the image classification task, each network learns with knowledge distilled from its peers in addition to traditional classification loss. However, the dynamics of learning in deep neural networks during training can be very complicated. Sometimes, learning can be slow, saturated, frozen, or even worse – dying out [15]. Hence, knowledge distilled from single steps/iterations can be chaotic or misleading. The rich information encoded in the learning trajectory is substantially ignored by previous OKD studies. To fill in this research gap, we propose to aggregate history information throughout the training process to build more superior teachers.

The proposed method consists of two components ensembling history information of logits and weights during the training process, respectively, as shown in Fig. 1. The proposed method potentially facilitates OKD by dampening the oscillations in the high curvature directions while speeding up the learning in the low curvature directions [38].

Authorized licensed use limited to: SUN YAT-SEN UNIVERSITY. Downloaded on November 27,2024 at 12:09:06 UTC from IEEE Xplore. Restrictions apply.

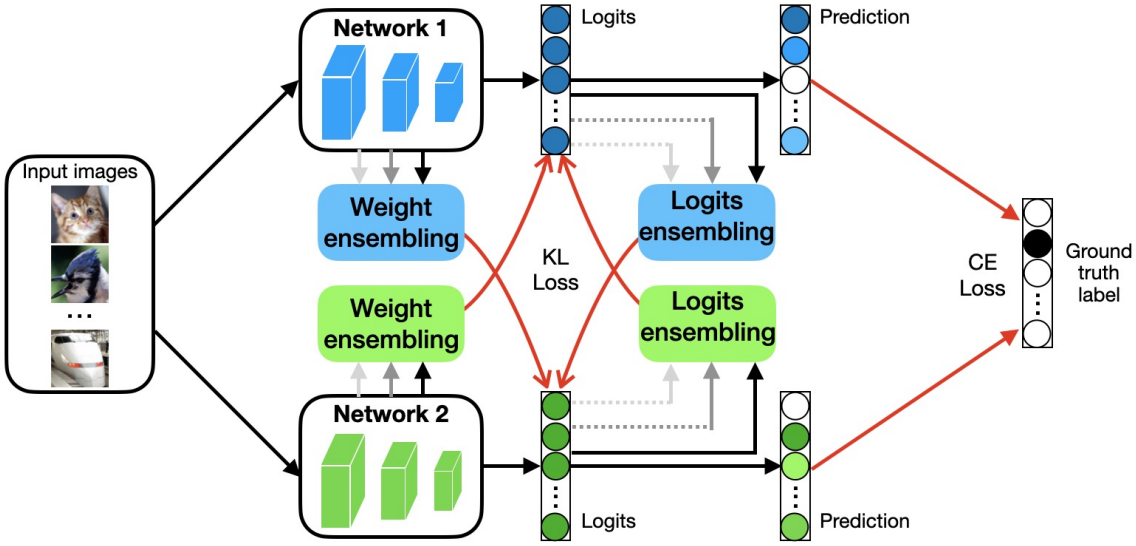


Fig. 1. Illustration of the proposed method with two student networks in the context of images classification. Each student network is associated with two modules for weight ensembling and logits ensembling, respectively. Lines directing to these two modules, with varying intensities, denote the current results and their historical values. Red lines indicate the calculation of losses, where KL loss and CE loss represent Kullback–Leibler divergence and cross-entropy loss, respectively.

B. Preliminaries

We first describe the basics of KD and OKD in the context of image classification. The general framework of the classical KD consists of a teacher network and a student network. The widely-used embodiment of knowledge transferred from the teacher network to the student network is the softened logits. Let T and S denote the teacher and the student model, W_t and W_s denote their parameters, respectively, and $X \in \mathcal{X}$ denote the training samples. The teacher's and the student's softened logits can be represented as in Eq. (1),

$$\begin{aligned} P_t^\tau &= T(X, W_t, \tau) = \text{softmax}(\frac{z_t}{\tau}), \\ P_s^\tau &= S(X, W_s, \tau) = \text{softmax}(\frac{z_s}{\tau}), \end{aligned} \tag{1}$$

where z_t and z_s are the logits of the teacher and the student model, respectively, and τ is the temperature that determines the level of logits softening. A higher τ results in more softened logits and the class probability distributed more evenly among all classes. Besides the commonly used cross-entropy loss, the student network also aims to mimic the softened logits output by the teacher model via Kullback–Leibler (KL) divergence loss as in Eq. (2),

$$\mathcal{L}_{KD} = \mathcal{L}_{CE}(P_s, y) + \mathcal{L}_{KL}(P_s^\tau, P_t^\tau), \tag{2}$$

where y is the ground truth label and P_s is the prediction of the student. $\mathcal{L}_{CE}$ and $\mathcal{L}_{KD}$ represent the cross-entropy loss and KL divergence loss, respectively.

In the context of OKD, all the models become each other's teachers while learning their own parameters. Without loss of generality, we suppose that there are two peer models in the OKD framework for convenience. Let M_i and P_i $(i = 1, 2)$ be the peer models and their corresponding logits, respectively. Then the objective function in OKD can be represented as in Eq. (3),

$$\mathcal{L}_{OKD} = \mathcal{L}_{CE}(P_i, y) + \mathcal{L}_{KL}(P_i^\tau, P_j^\tau), \tag{3}$$

where $i, j = 1, 2$ and $i \neq j$. In the OKD scenario, lower temperatures usually exhibit better performance. As a result, the temperature τ is commonly set to 1. Hence, in the following context, we omit the superscript τ for simplicity.

C. Logits Ensembling

In order to extract superior knowledge from one peer for others, we propose two modules associated with each student network at different levels to incorporate knowledge from the learning trajectories in the OKD procedure, i.e, logits-level ensembling and weight-level ensembling.

The first module is referred to as the logits ensembling, which incorporates the current logits and the historical values to produce an enhanced target. Denote P_i^t the output logits of model i at the t-th iteration. We leverage an LSTM network that takes the logits sequence of the past L time steps $\mathbf{P}_i^t = [P_i^{t-(L-1)}, P_i^{t-(L-2)}, \cdots, P_i^{t-1}, P_i^t]$ as the input, and generates an integrated target $\hat{P}_i^t$, which is presented in Eq. (4). The procedure of the LSTM network is demonstrated in Fig. 2.

$$\hat{P}_i^t = \text{LSTM}(\mathbf{P}_i^t). \tag{4}$$

Instead of learning from P^t, each model learns from the target $\hat{P}^t$ that encodes the temporal trajectory of the logits of others peer as in Eq. (5),

$$\mathcal{L}_{LE} = \mathcal{L}_{KL}(P_i^t, \hat{P}_j^t), \tag{5}$$

Authorized licensed use limited to: SUN YAT-SEN UNIVERSITY. Downloaded on November 27,2024 at 12:09:06 UTC from IEEE Xplore. Restrictions apply.

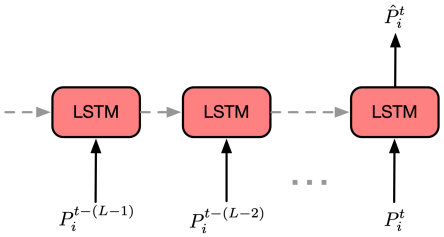


Fig. 2. The logits ensembling module. A classic LSTM [39] network takes the logits of a model at the previous $L-1$ time steps and the current step as the input, and outputs an integrated target, which is used for other peers' to learn towards. Solid arrows indicate the input and output; dashed arrows indicate the hidden state.

where $i \neq j$.

For the training of the LSTM network, we use the ground truth label as the target and cross-entropy as the objective function. As a supplement, we also introduce the average logits of the peers at the current time step t, i.e., $\frac{1}{N}\sum_{i=1}^{N} P_i^t$ (where N is the number of peer models in the system), as an extra hint. Eq. (6) shows a scenario where $N = 2$.

$$\mathcal{L}_{LSTM} = \mathcal{L}_{CE}(\hat{P}_i^t, y) + \mathcal{L}_{KL}(\hat{P}_i^t, \frac{1}{2}(P_i^t + P_j^t)). \quad (6)$$

The LSTM network can be considered as encoding the logits-level learning trajectory and becoming a stronger teacher while regulated by the ground truth label.

D. Weight Ensembling

Besides the logits-level ensembling, we also propose to utilize the history information of each model's weight to provide a further alignment among the peers. Inspired by the previous weight averaging work in the semi-supervised domain [21], we establish a surrogate model that has the identical structure as that of each student network, whose parameters are updated via the exponential moving average of the student's weights at each time step during the OKD training procedure. Denote $\bar{W}_i^t$ the parameters of the surrogate model $\bar{M}_i^t$ associated with the student network M_i^t. Then $\bar{W}_i^t$ is periodically updated with Eq. (7),

$$\bar{W}_i^t = \lambda_w \bar{W}_i^{t-1} + (1-\lambda_w) W_i^t, \quad (7)$$

where λ_w, referred to as the EMA decay, can be viewed as a smoothing term that controls how much history data is involved. Therefore, the output of the surrogate model can be represented as

$$\bar{P}_i^t = \bar{M}_i^t(X, \bar{W}_i^t), \quad (8)$$

and the objective function for the weight ensembling module can be written as in Eq. (9),

$$\mathcal{L}_{WE} = \mathcal{L}_{KL}(P_i^t, \bar{P}_j^t), \quad (9)$$

where $i \neq j$.

Finally, the overall loss function of our proposed approach can be represented as in Eq. (10),

$$\mathcal{L} = \mathcal{L}_{CE}(P_i^t, y) + \lambda_{le}\mathcal{L}_{LE} + \lambda_{we}\mathcal{L}_{WE}, \quad (10)$$

where λ_{le} and λ_{we} are the two scaling factors that balance the importance of the logits-level and the weight-level modules, respectively, compared to the cross-entropy loss.

IV. Experiments

In this section, we evaluate the proposed method with experiments on various benchmark datasets and structures. Then we conduct several ablation studies on the effectiveness of two proposed modules and the effect on varying their key hyperparameters.

A. Setup

1) Datasets and Network Structures: We conduct several experiments to evaluate the proposed method using a wide range of benchmark network structures (including ResNets, WRNs, VGG, and MobileNet) on various datasets (including CIFAR-10, CIFAR-100, and ImageNet) for the task of image classification. CIFAR-10 and CIFAR-100 [40] contain 60000 RGB images with a resolution of 32×32 out of 10 and 100 classes, respectively. These images are spilt into $50,000$ as the training set and $10,000$ as the test set. We pad the training samples with two pixels on each side and then implement a random crop that resizes it back to the original size. A random horizontal flip is also used for augmentation purposes. ImageNet (ILSVRC 2012) [41] is a large-scale object classification dataset with 1000 categories and 1.2 million images. For the training samples, we randomly crop the images and then resize them to the resolution of 224×224. After that, a random horizontal flip is also implemented. For the test samples, they are center-cropped to a resolution of 224×224. We use the standard implementation of ResNet-20, ResNet-32 [42], WRN-16-4, WRN-28-2 [43], VGG-13 [44] and MobileNet [45] as the peers in the proposed OKD approach.

2) Compared Methods: We use independent training (training the models solely with the cross-entropy loss) and Deep Mutual Learning (DML) [28] as two baselines. In addition, we compare the proposed method with several state-of-the-art OKD methods, including KDCL [12], ONE [14], and OKDDip [13]. Task-specific hyperparameters are described in related sections.

B. Performance Comparison with CIFAR-10/100

We first evaluate our proposed approach with the CIFAR-10 and CIFAR-100 datasets. We consider two scenarios, i.e., the peer networks in the OKD training are with (1) the same and (2) different structures. For all the experiments, we train each peer model for 240 epochs with a batch size of 64 and an SGD optimizer (the momentum is set to 0.9). The learning rate is initialized to 0.01 for MobileNet and 0.05 for others, which is decayed by a factor of 10 at the 150th, 180th, and 210th epochs. The weight decay factor is set to 0.0005. We

Authorized licensed use limited to: SUN YAT-SEN UNIVERSITY. Downloaded on November 27,2024 at 12:09:06 UTC from IEEE Xplore. Restrictions apply.

TABLE I
TOP-1 ERROR RATES (%) ON CIFAR-10 AND CIFAR-100 WITH SAME PEER STRUCTURES. THE FIRST AND SECOND COLUMNS OF EACH STRUCTURE REPRESENT THE AVERAGE AND THE ENSEMBLE PERFORMANCE OF TWO NETWORKS, RESPECTIVELY.

Datasets	Methods	ResNet-20	ResNet-32	WRN-16-4	WRN-28-2	VGG13	MobileNet
CIFAR-10	Ind	8.11/-	7.23/-	5.43/-	5.66/-	9.42/-	12.47/-
	DML	7.96/7.37	6.90/6.32	5.04/4.65	5.25/4.84	9.45/8.87	12.08/11.50
	ONE	7.37/6.79	6.21/5.81	4.55/4.39	5.16/4.76	7.18/7.17	10.81/10.80
	KDCL	7.29/6.82	6.37/5.53	4.90/4.55	4.97/4.47	6.52/6.05	10.11/8.96
	OKDDip	7.04/6.79	6.20/5.79	5.76/5.70	5.00/4.91	5.85/5.84	-/-
	Ours	**6.71/6.03**	**5.85/5.42**	**4.39/4.09**	**4.91/4.17**	**5.41/4.76**	**9.78/8.86**
CIFAR-100	Ind	32.52/ -	31.01/-	24.62/-	26.50/-	25.36/-	35.40/-
	DML	31.08/29.29	28.83/27.02	23.87/22.34	24.64/22.77	26.38/25.01	36.54/34.72
	ONE	29.48/27.79	27.31/25.59	22.59/21.60	23.26/21.58	24.51/23.89	36.68/36.73
	KDCL	29.53/27.42	27.31/25.58	21.04/**19.49**	22.75/**20.48**	26.05/24.66	33.90/31.53
	OKDDip	29.60/28.08	26.89/25.18	23.60/22.53	23.31/21.63	24.82/24.77	-/-
	Ours	**28.91/27.31**	**26.69/24.82**	**20.48**/19.53	**22.28**/20.72	**24.01/21.66**	**32.90/31.51**

stick to the same set of the above hyperparameters for the implementation of the compared approaches.

For the logits ensembling module, we employ a two-layer LSTM network with the number of features in the hidden state set to 20. We use a sequence of the last 10 logits produced by each peer as the input of the module. The learning rate for training the LSTM network is set to 0.05. We tried to decay this learning rate inconsistent with the peer model's learning rate and it resulted in similar performance. Therefore, we use the constant learning rate across the experiments. For the weight ensembling surrogate model, we set the momentum coefficient λ_w to 0.5, which is proved to be the most effective selection among the massive empirical studies (details presented in the Ablation Study section IV-D). We warm up the learning procedure for 15 epochs by training each peer independently with the cross-entropy loss, which provides a proper initial path of the learning trajectory. It is worth mentioning that both the logits ensembling and weight ensembling can be updated on the fly during the training procedure with constant overhead.

1) Evaluation with Same-structure Peers: We start with the evaluation using two peers with the same structure. The experimental results are presented in Table I. Both the average and the ensemble performance of two networks in terms of top-1 error rates (%) are reported. Our proposed approach outperforms all the compared methods except for the accuracy of the ensemble of two cases with WRNs on CIFAR-100. In most cases, using the proposed method with two trajectory-involved modules in the OKD procedure results in a substantial decrease in top-1 errors on both the average of individual peers and the ensemble of them. The proposed approach generally lowers the error rates (%) by the value in the range of from 0.3 to 0.4 over existing methods. It is also observed that the performance improvement of model ensemble against individual models with our approach is higher than others. These results indicate that peers learn more diversified information by incorporating historical information.

2) Evaluation with Different-structure Peers: We also evaluate the proposed approach with the scenario where the peers are with different architectures. We use the structures in Table I and five different combinations for pairs in this experiment. Since branch-based approaches (ONE and OKDDip[1]) cannot deal with the situation in which network structures are different, we only compare with DML and KDCL.

The experimental results are shown in Table II. We also report the averaged performance of the two networks and the performance of their ensemble on top-1 error rates (%). Similar to the experiments with the same structure peers, the proposed approach exhibit better performance compared to DML and KDCL. It is worth noting that the performance of the models with smaller capacity among the peers is significantly improved. For example, when a ResNet-32 learns from a WRN-16-4, an error rate of 25.66% is achieved, which is 1.03% better than learning from another ResNet-32 model and is 1.06% better than the state-of-the-art, i.e., KDCL. These results demonstrate that (1) the learning trajectories of the models with larger capacity indeed encode plentiful knowledge and (2) the proposed approach effectively catches and leverages this information. We also notice that KDCL achieves quite satisfactory performance in our evaluations. This might be because that although KDCL is an OKD approach, it makes a great effort in manipulating the training samples by adding noises for the adversarial training purpose. Without this kind of manipulation, our approach still achieves competitive performance.

C. Performance Comparison with ImageNet

To evaluate the effectiveness of the proposed approach on large-scale tasks, we take ResNet-34 on the ImageNet dataset as the experiment. Here we use two pre-trained ResNet-34 models as the initial starting point. We take the pre-trained ResNet-34 from the Torchvision package as one model, and the other is trained from scratch for 90 epochs, with a learning rate starting from 0.01 and divided by 10 every 30 epochs. The average performance of the two pre-trained models in terms of the top-1 error rate is 26.69%, which is considered as the baseline. Other hyperparameters remain the

[1]OKDDip can be extended to a network-based version but the source code is not provided by the authors. So we do not compare to OKDDip here.

Authorized licensed use limited to: SUN YAT-SEN UNIVERSITY. Downloaded on November 27,2024 at 12:09:06 UTC from IEEE Xplore. Restrictions apply.

TABLE II

TOP-1 ERROR RATES (%) ON CIFAR-10 AND CIFAR-100 WITH DIFFERENT PEER STRUCTURES. THE THREE NUMBERS IN EACH COMBINATION REPRESENT THE PERFORMANCE OF THE FIRST NETWORK, THE SECOND NETWORK, AND THE ENSEMBLE OF TWO NETWORKS, RESPECTIVELY.

Datasets	Methods	ResNet-20 ResNet-32	ResNet-32 WRN-16-4	WRN-16-4 WRN-28-2	MobileNet WRN-16-4	VGG13 WRN-16-2
CIFAR-10	Ind	8.11/7.23/-	7.23/5.43/-	5.43/5.66/-	12.47/5.43/-	9.42/6.79/-
	DML	7.84/7.37/6.79	6.78/5.45/5.49	5.07/5.50/4.92	11.15/6.95/8.10	7.12/6.94/6.34
	KDCL	6.93/6.47/6.16	6.28/4.92/5.12	4.83/5.22/4.70	9.89/5.61/6.62	6.29/6.59/5.67
	Ours	**6.51/6.04/5.99**	**5.95/4.41/4.76**	**4.29/4.38/3.96**	**9.76/4.77/5.76**	**5.37/5.39/4.92**
CIFAR-100	Ind	32.52/31.01/-	31.01/24.62/-	24.62/26.50/-	35.40/24.62/-	25.36/26.74/-
	DML	31.69/29.72/28.74	28.10/23.39/24.07	23.06/24.67/22.04	36.16/27.13/28.63	28.10/28.70/25.76
	KDCL	30.10/27.01/26.03	26.72/21.30/**21.21**	21.36/23.24/20.02	32.67/**22.50**/24.08	23.58/25.89/25.39
	Ours	**28.55/26.69/25.44**	**25.66/21.26**/21.62	**20.56/22.20/19.87**	**30.60**/22.77/**23.09**	**22.64/24.64/22.23**

TABLE III

TOP-1 ERROR RATES (%) ON IMAGENET WITH RESNET-34. THE REPORTED NUMBERS ARE THE AVERAGE ERROR RATES OF TWO MODELS. THE ERROR RATES OF THE DML AND THE ONE METHODS ARE FROM [13].

Datasets	Network Types	Ind	DML	ONE	OKDDip	Ours
ImageNet	ResNet-34	26.69	26.03	25.92	25.60	**25.18**

same as the previous experiments. From the experiment results (Table IV-B2) we can observe that our proposed approach outperforms other recent OKD approaches. Specifically, the top-1 error rate decreases by 1.51% and 0.85% compared to the baseline and DML, respectively. Compared to the state-of-the-art approach, i.e., OKDDip, we achieve a performance improvement of 0.42% in terms of top-1 test accuracy. These results validate the effectiveness of our proposed approach on the large-scale object classification dataset.

D. Ablation Study

In this section, we conduct several groups of ablation studies to illustrate the effectiveness of each principal component of the proposed approach, as well as the selection of crucial hyperparameters (lengths of history information and the importance of weight-level learning trajectory). We employ ResNet-32 and WRN-28-2 with CIFAR-10 and CIFAR-100 for the ablation studies.

1) Effect of Different Components: We first test if the two proposed history-aware modules have any effect on the performance of peer networks with the following ablation experiments: (1) mutually training two peers without any of the two proposed modules, i.e., training with the DML scheme; (2) training with the weight-level ensembling module only, i.e., the temporal mean surrogate model; (3) training with the logits-level ensembling module only, i.e., the LSTM network; and (4) training with both modules. Experiment results are presented in Table IV, which validate that employing either of the two modules substantially improves the models' performance over DML with a clear margin. When both modules are implemented, the performance of both networks on both datasets is further improved as expected.

In addition, we observe that the weight-level ensembling module outperforms the logits-level module when only one of the modules is used. For example, on the CIFAR-100 dataset, when the weight-level module is used, the top-1 test errors drop by 2.02% and 1.70% with ResNet-32 and WRN-28-2, respectively. In comparison, the drops are only 0.98% and 1.21% when only the logits-level module is used. These results indicate that the weight-level ensembling can involve more useful information through averaging the whole parameters of the network at each step of the training.

TABLE IV

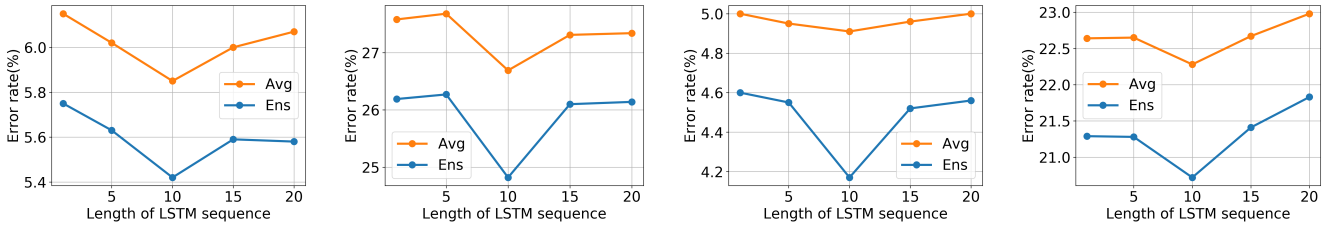
TOP-1 ERROR RATES (%) ON CIFAR-10 AND CIFAR-100 WITH RESNET-32 AND WRN-28-2 WITH DIFFERENT COMPONENTS OF THE PROPOSED METHOD. WE: WEIGHT ENSEMBLING, LE: LOGITS ENSEMBLING. THE FIRST AND SECOND COLUMNS OF EACH EXPERIMENT REPRESENT THE AVERAGE AND THE ENSEMBLE PERFORMANCE OF TWO NETWORKS, RESPECTIVELY.

Datasets	Component WE	 LE	ResNet-32	WRN-28-2
CIFAR-10			6.90/6.32	5.25/4.84
		✓	6.04/5.54	5.02/4.38
	✓		6.12/5.78	4.99/4.53
	✓	✓	5.85/5.42	4.91/4.17
CIFAR-100			28.83/27.02	24.64/22.77
		✓	27.85/25.82	23.43/21.49
	✓		26.81/25.88	22.94/20.89
	✓	✓	26.69/24.82	22.28/20.72

2) Effect of the Length of History Information: We continue to investigate whether varying the length of the history information, i.e., the input sequences to the LSTM network impacts the performance of the peer models. We continue to use the same network structures (ResNet-32 and WRN-28-2) and datasets (CIFAR-10 and CIFAR-100) for this ablation study. We try different lengths of history information used in the logits ensembling module, i.e., the lengths of 1, 5, 10, 15, and 20, respectively. All the experiments are conducted with 3 trials and the averaged results are presented in Fig. 3.

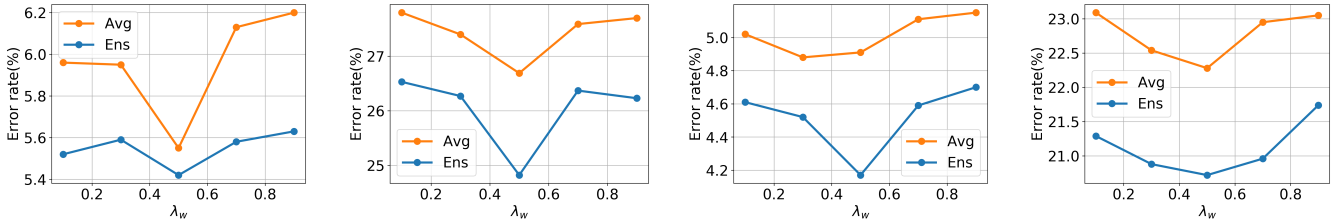
With these experiments, we observe that with a sequence length of 10, the models in all the four experiments achieve the best performance in terms of not only the average error rates of the two peer models but also the ensemble of them. When the input sequence length gets smaller or larger, the performance gradually degrades. These results are intuitively reasonable

Authorized licensed use limited to: SUN YAT-SEN UNIVERSITY. Downloaded on November 27,2024 at 12:09:06 UTC from IEEE Xplore. Restrictions apply.



(a) ResNet-32, CIFAR-10 (b) ResNet-32, CIFAR-100 (c) WRN-28-2, CIFAR-10 (d) WRN-28-2, CIFAR-100

Fig. 3. Top-1 error rates (%) on the CIFAR-10 and CIFAR-100 datasets with the ResNet-32 and the WRN-28-2 structures with different lengths of input sequences of the logits-level ensembling module. Avg and Ens indicate the average error rates of the two models and the error rate of the ensemble of the two models, respectively.



(a) ResNet-32, CIFAR-10 (b) ResNet-32, CIFAR-100 (c) WRN-28-2, CIFAR-10 (d) WRN-28-2, CIFAR-100

Fig. 4. Top-1 error rates (%) on the CIFAR-10 and CIFAR-100 datasets with the ResNet-32 and the WRN-28-2 structures with different momentum coefficients λ_w of the weight-level ensembling module. Avg and Ens indicate the average error rates of the two models and the error rate of the ensemble of the two models, respectively.

because an extremely short sequence length encodes very little information of the learning trajectory and the performance regresses to that as using DML. On the other hand, an extremely long sequence is somewhat overstaffed, considering outdated outputs that have been deviated too much from the current stage as useful information. Besides, using longer sequences as the input makes the training procedure more time-consuming and introduces extra difficulties to the training of LSTM networks. In the context of the experiments, using an input length of 10 is an effective choice in terms of both performance and efficiency.

3) How long does the weight-level learning trajectory need to be?: Another essential hyperparameter is the EMA decay λ_w in the weight ensembling, which determines how much history data is involved in the averaging. To evaluate the sensitivity of the weight-level module to the λ_w, we vary its values in the range of 0.1, 0.3, 0.5, 0.7, 0.9, and report the corresponding performance of the peer models. ResNet-32 and WRN-28-2 on CIFAR-10 and CIFAR-100 as in previous ablation studies are used for the evaluation.

As shown in Fig. 4, when $\lambda_w = 0.5$, the most superior performance is achieved nearly in all cases, except for the case of WRN-28-2 on CIFAR-10 in which using $\lambda_w = 0.3$ leads to a performance gain of 0.03% in terms of test accuracy. These results indicate that, approximately, only the last two or three time steps (i.e., $\lambda_w = 0.5$) in the learning trajectory dominate the hidden information during the learning process. This is quite surprising because we expected this number to be much larger (say, using ten-time steps with $\lambda_w = 0.9$). One of the potential explanations is that, compared to the logits sequences, the peer model's parameters change much more dynamically, and averaging within a wide temporal window size smooths out important information. Another hypothesis is that the weight-level trajectory is too complicated for a straightforward momentum ensembling to handle with. The momentum ensembling pre-defines the integration scheme without considering the dynamic learning process. With these concerns, our immediate future work will focus on investigating the possibility of leveraging more advanced encoding strategies (for instance, designing an integration network as implemented in the logits-level module) for the online mutual learning of feature-level trajectories.

V. Conclusion

In this study, we proposed to leverage the learning trajectory during the training process of each peer model in the online knowledge distillation framework as an extra alignment to guide the models towards stable and superior performance. Two ensembling modules served as enhanced instructors, were designed to encode the hidden information produced during the online knowledge distillation training procedure at distinct levels. Specifically, an LSTM network was employed to integrate the temporal logits-level paths, while a surrogate model associated with each peer was utilized to ensemble the weight-level learning trajectory. Instead of learning mutually from each peer model's outputs at a single time step/iteration, learning from peers' learning trajectories at different levels proved as a more effective paradigm with a variety of widely used network structures and benchmark object classification datasets.

Authorized licensed use limited to: SUN YAT-SEN UNIVERSITY. Downloaded on November 27,2024 at 12:09:06 UTC from IEEE Xplore. Restrictions apply.

REFERENCES

[1] Zhenpeng Chen, Yanbin Cao, Yuanqiang Liu, Haoyu Wang, Tao Xie, and Xuanzhe Liu. A comprehensive study on challenges in deploying deep learning based software. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 750–762, 2020.

[2] Chengcheng Li, Zi Wang, and Hairong Qi. Fast-converging conditional generative adversarial networks for image synthesis. In *2018 25th IEEE international conference on image processing (ICIP)*, pages 2132–2136. IEEE, 2018.

[3] Zhewei Yao, Zhen Dong, Zhangcheng Zheng, Amir Gholami, Jiali Yu, Eric Tan, Leyuan Wang, Qijing Huang, Yida Wang, Michael Mahoney, et al. Hawq-v3: Dyadic neural network quantization. In *International Conference on Machine Learning*, pages 11875–11886. PMLR, 2021.

[4] Yehui Tang, Yunhe Wang, Yixing Xu, Dacheng Tao, Chunjing Xu, Chao Xu, and Chang Xu. Scop: Scientific control for reliable neural network pruning. *arXiv preprint arXiv:2010.10732*, 2020.

[5] Zi Wang, Chengcheng Li, Xiangyang Wang, and Dali Wang. Towards efficient convolutional neural networks through low-error filter saliency estimation. In *Pacific Rim International Conference on Artificial Intelligence*, pages 255–267. Springer, 2019.

[6] Chengcheng Li, Zi Wang, and Hairong Qi. An efficient pipeline for pruning convolutional neural networks. In *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 907–912. IEEE, 2020.

[7] Zi Wang, Chengcheng Li, and Xiangyang Wang. Convolutional neural network pruning with structural redundancy reduction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14913–14922, 2021.

[8] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pages 116–131, 2018.

[9] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, 2021.

[10] Guodong Xu, Ziwei Liu, Xiaoxiao Li, and Chen Change Loy. Knowledge distillation meets self-supervision. In *European Conference on Computer Vision*, pages 588–604. Springer, 2020.

[11] Zi Wang. Data-free knowledge distillation with soft targeted transfer set synthesis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10245–10253, 2021.

[12] 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.

[13] Defang Chen, Jian-Ping Mei, Can Wang, Yan Feng, and Chun Chen. Online knowledge distillation with diverse peers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3430–3437, 2020.

[14] Xu Lan, Xiatian Zhu, and Shaogang Gong. Knowledge distillation by on-the-fly native ensemble. *arXiv preprint arXiv:1806.04606*, 2018.

[15] Remi Tachet des Combes, Mohammad Pezeshki, Samira Shabanian, Aaron Courville, and Yoshua Bengio. On the learning dynamics of deep neural networks. *arXiv preprint arXiv:1809.06848*, 1(2.1):3, 2018.

[16] Wei Wu, Xiaoyuan Jing, Wencai Du, and Guoliang Chen. Learning dynamics of gradient descent optimization in deep neural networks. *Science China Information Sciences*, 64(5):1–15, 2021.

[17] Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4794–4802, 2019.

[18] Larry R Medsker and LC Jain. Recurrent neural networks. *Design and Applications*, 5:64–67, 2001.

[19] Boris T Polyak and Anatoli B Juditsky. Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 30(4):838–855, 1992.

[20] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

[21] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *arXiv preprint arXiv:1703.01780*, 2017.

[22] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541, 2006.

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

[24] Chengchao Shen, Xinchao Wang, Youtan Yin, Jie Song, Sihui Luo, and Mingli Song. Progressive network grafting for few-shot knowledge distillation. *arXiv preprint arXiv:2012.04915*, 2020.

[25] Zi Wang. Zero-shot knowledge distillation from a decision-based black-box model. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 10675–10685. PMLR, 18–24 Jul 2021.

[26] Fei Yuan, Linjun Shou, Jian Pei, Wutao Lin, Ming Gong, Yan Fu, and Daxin Jiang. Reinforced multi-teacher selection for knowledge distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14284–14291, 2021.

[27] Lin Wang and Kuk-Jin Yoon. Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[28] 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.

[29] Inseop Chung, SeongUk Park, Jangho Kim, and Nojun Kwak. Feature-map-level online adversarial knowledge distillation. In *International Conference on Machine Learning*, pages 2006–2015. PMLR, 2020.

[30] Chengcheng Li, Zi Wang, and Hairong Qi. Online knowledge distillation by temporal-spatial boosting. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 197–206, 2022.

[31] Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1365–1374, 2019.

[32] SeongUk Park and Nojun Kwak. Feed: Feature-level ensemble for knowledge distillation. *arXiv preprint arXiv:1909.10754*, 2019.

[33] Mingi Ji, Byeongho Heo, and Sungrae Park. Show, attend and distill: Knowledge distillation via attention-based feature matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7945–7952, 2021.

[34] Gustavo Aguilar, Yuan Ling, Yu Zhang, Benjamin Yao, Xing Fan, and Chenlei Guo. Knowledge distillation from internal representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7350–7357, 2020.

[35] Guile Wu and Shaogang Gong. Peer collaborative learning for online knowledge distillation. In *AAAI*, 2021.

[36] Shawn Beaulieu, Lapo Frati, Thomas Miconi, Joel Lehman, Kenneth O Stanley, Jeff Clune, and Nick Cheney. Learning to continually learn. *arXiv preprint arXiv:2002.09571*, 2020.

[37] Sebastian Flennerhag, Pablo Garcia Moreno, Neil Lawrence, and Andreas Damianou. Transferring knowledge across learning processes. In *International Conference on Learning Representations*, 2019.

[38] Michael R Zhang, James Lucas, Geoffrey Hinton, and Jimmy Ba. Lookahead optimizer: k steps forward, 1 step back. *arXiv preprint arXiv:1907.08610*, 2019.

[39] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

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

[41] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[42] 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*, pages 770–778, 2016.

[43] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

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

[45] 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.

Authorized licensed use limited to: SUN YAT-SEN UNIVERSITY. Downloaded on November 27,2024 at 12:09:06 UTC from IEEE Xplore. Restrictions apply.