Research on Knowledge Distillation Regularization Methods*

WANG Xuechun

(*School of Mathematics and System Sciences, Xinjiang University, Urumqi Xinjiang 830017, China*)

Abstract: In deep learning, regularization is extremely important as it prevents overfitting models and improves their generalization performances. A relatively new, yet increasingly popular type of regularization is knowledge distillation (KD), a set of techniques for soft labels generated by one model as supervised signals to guide the training of another model. We first explain the fundamentals of KD regularization and then categorize KD regularization strategies into two different types, viz. forward distillation and mutual distillation. For each type, we discuss in detail its key components and representative methods. After comparing the pros and cons of KD regularization strategies and testing their performance on the common benchmark of image classification, we provide guidelines on how to choose appropriate KD regularization techniques for specific scenarios. Finally, we identify a number of key challenges and discuss future research directions of KD regularization.

Key words: knowledge distillation; model generalization; overfitting; regularization.

DOI: 10.13568/j.cnki.651094.651316.2023.02.26.0002

CLC number: TP181 **Document Code**: A **Article ID**: 2096-7675(2023)05-0534-09

引文格式：王雪纯. 知识蒸馏正则化方法研究[J]. 新疆大学学报(自然科学版)(中英文), 2023, 40(5): 534-542+549.

英文引文格式：WANG Xuechun. Research on knowledge distillation regularization methods[J]. Journal of Xinjiang University(Natural Science Edition in Chinese and English), 2023, 40(5): 534-542+549.

知识蒸馏正则化方法研究

王 雪 纯

(新疆大学 数学与系统科学学院，新疆 乌鲁木齐 830017)

摘 要： 在深度学习中, 正则化是防止模型过拟合和提高模型泛化性能的重要工具. 知识蒸馏（Knowledge Distillation, KD）是一组由一个模型生成的软标签作为监督信号去指导另一个模型的相对较新的, 流行的正则化方法. 首先, 给出了KD正则化的基本知识并将现有的知识蒸馏正则化分为两大类, 即正向蒸馏和互蒸馏. 对每种类型, 都详细介绍了关键的组成部分和代表性方法. 其次, 比较了这两大类正则化方法的优缺点并在图像分类上评估了模型的泛化性能. 同时, 也为特定的任务和场景选择合适的KD正则化方法提供了指南. 最后, 总结了KD正则化方法存在的关键性挑战并讨论了将来的研究方向.

关键词： 知识蒸馏; 模型泛化; 过拟合; 正则化

0 Introduction

Deep learning has been tremendously successful in computer vision[1], natural language processing[2], and many other fields. Large pre-trained models, such as the GPT family and BERT, frequently exhibit excellent performance, but have millions and even billions of learnable parameters that far exceed the number of training samples. In this case, a foundation model tends to suffer from overfitting, which is essentially the mistaking of certain residual variations such as noises for the true structure of the problem. To prevent this from happening, it is necessary to substantially reduce generalization errors on

* **Received Date**: 2023-02-26

Foundation Item: This work was supported by subproject of the Regional Innovation Joint Fund "Theory and methodology of reasonable personalized recommendation based on graph neural networks" (U19A2079).

Biography: WANG Xuechun (1993-), female, master student, research fields: deep learning, E-mail: wxcmath@163.com.

unseen data without excessively increasing errors on the training data. Naturally, overfitting can be avoided by increasing the amount of training data. Another popular technique is regularization, which limits the complexity of the model yet improves its prediction performance on unseen data. Various canonical regularization techniques exist, such as weight decay families[3−4], dropout[5], normalization[6−7], and data augmentation families[8−10].

The conventional regularization methods such as L_1 and L_2 commonly reduce the complexity of the model by adding extra regularization terms to the original loss function based on its parameters. In contrast, KD regularization is a technique that incorporates the concept of KD into the regularizer term of the loss function. In addition to the standard loss term, an extra regularization term is added to encourage the student model's output to match the soft targets of the teacher model. This regularization term acts as a regularizer during training and helps the student model capture the knowledge contained in the teacher model.

KD and KD regularization are both techniques used to transfer knowledge from a larger, more complex model to a smaller, simpler model. However, there is a difference in the way they are applied. KD is a standalone approach where a student model is trained to mimic a teacher model through the use of soft targets, while KD regularization integrates the principles of KD into the regularization term of the loss function to guide the training of the student model. Early studies have demonstrated the effectiveness of KD in model compression[11−12] and semi-supervised learning[13], but ignored the prospect that KD also acts as a regularizer to enhance the generalizability of models. Ba et al.[14] and Hinton et al.[15] demonstrated the soft labels generated by the teacher model can effectively regularize the training of the student models and alleviate overfitting. Subsequently, scholars have delved into various KD regularization techniques to improve generalization performance of the deep model such as [16-21], but these fragmented regularization techniques do not provide a systematic introduction for researchers. For this reason, our contribution is to provide a relatively comprehensive taxonomy and exploration of KD regularization strategies for achieving model generalization.

In the literature, there already exist excellent surveys on regularization. For example, Moradi et al.[22] inspected the most effective regularization techniques and their variants without mentioning any applications. Sanyos et al.[23] discussed regularization for convolutional neural networks. Tang et al.[24] presented sparse regularization techniques from the perspective of model compression. Tian et al.[25] reviewed a wide range of regularization methods and applications, but they did not cover KD regularization.

To the best of our knowledge, there are no comprehensive surveys devoted exclusively to KD regularization. Yet the rapid advances in research on this topic call for review articles that explain the fundamental ideas, classify the variants into simple and exhaustive categories, compare the pros and cons of different strategies, and provide guidelines for application scientists to choose appropriate KD regularization methods for specific tasks.

This paper is a survey that answers the aforementioned demands. In section 1, we lay the foundation by explaining basic definitions, fundamental principles, and key mechanisms of the KD regularization. The main part of this survey consists of sections 2 and 3, where we discuss in detail the two main KD regularization strategies, namely forward distillation, mutual distillation. In section 4, we summarize some key points of our review into table, which hopefully could serve as a road map for this seemingly chaotic field of KD regularization schemes. In particular, we benchmark different KD regularization strategies by the practical problem of image classification. In section 5, we discuss the limitations of current KD regularization techniques and give some corresponding research prospects.

1 Fundamentals

1.1 The definition of KD regularization

In general, knowledge refers to the awareness of facts or practical skills. In this context, knowledge includes in a broad sense everything that can be utilized by other models such as parameters, features, structures, modules, and so on. In a narrow sense, knowledge is the output of a teacher model that can be utilized in training other models.

In chemistry, distillation is a method for separating components with different boiling points by heating the compound solution. In the context of this paper, distillation is the process of obtaining pure knowledge from impure knowledge by amplifying the similarity of knowledge across different models. The name "distillation" comes from the analogy to chemical distillations that (i) during the training phase the knowledge similarity is amplified by increasing the value of a parameter (the so-called "temperature") and (ii) during the testing phase the temperature is lowered to extract knowledge from the original model.

Table 1 The difference between KD and transfer learning

	KD	Transfer learning
Data domains	Transfers on the same target datasets	Transfers on different target datasets
Architectures	Big and small networks	A single network
Learning styles	Does not directly use learned weights	Uses weights from rich data in other domains
Main purposes	Approximates large networks with small networks	Learns new knowledge from existing knowledge

KD is a teacher-student training process in which a lightweight student model is trained by extracting knowledge from the output of a sizable mature teacher model. Given the knowledge from the pre-trained teacher model, these students are supposed to be competitive to or even superior to the teacher. KD is similar to transfer learning[26] in that, both of them involve certain transfer processes, but KD emphasizes knowledge transfer over weight transfer. See Table 1 for a comparison of the two concepts in terms of data domains, architectures, learning styles, and main purposes.

1.2 The training process in KD regularization

The prediction of a category probability usually benefits from the output logits of the teacher model passed through a softmax layer. As an early work, Ba and Caruana[14] trained a small network by learning logit outputs from a big network. However, when the probability distribution output by softmax is employed directly, the probabilities in the negative labels are very much flattened to zero, resulting in an information loss. To resolve these issues, Hinton et al.[15] added a temperature parameter T to the original softmax function to reduce the difference between target and non-target categories so that the information contained in the negative labels is amplified. The softened class probability is calculated as

$$q_i = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)} \tag{1}$$

where z_i, z_j are input logits of the softmax function and T is the temperature that controls the softening of the output probability. For $T = 1$, the soften softmax function reduces to the standard softmax function. As T increases, the class probability distribution becomes smoother and smoother.

In particular, the softened knowledge of the teacher model is employed as the first portion of the loss function to regulate the parameters of the student model, which is trained with both hard and soft targets. The distillation loss $\mathcal{L}_{\text{soft}}$ is composed of a teacher model and a student model, both of which have $T > 1$, while the student loss $\mathcal{L}_{\text{hard}}$ only concerns the student model with temperature parameter $T = 1$ and ground truth label. The entire loss function is then

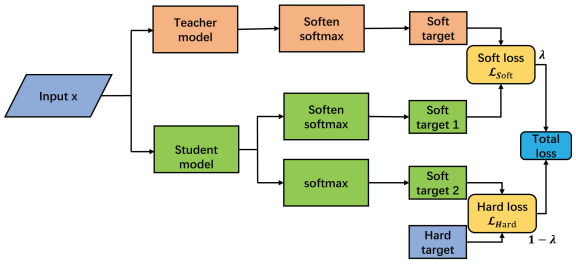

Fig 1 The training process of KD regularization[15]

$$\mathcal{L} = \lambda \mathcal{L}_{\text{soft}} + (1-\lambda)\mathcal{L}_{\text{hard}} \tag{2}$$

where λ is the balance factor for the soft and hard targets, the distillation loss is

$$\mathcal{L}_{\text{soft}}(p^{(T)}, q^{(T)}) = -\sum_i p_i^T \log(q_i^{(T)}) \tag{3}$$

and the student loss is

$$\mathcal{L}_{\text{hard}}(y, q^{(1)}) = -\sum_i y_i \log(q_i^{(1)}) \tag{4}$$

where y_i is a one-hot encoding. See Fig 1 for an illustration of the training framework of KD.

Thanks to the soft targets, KD is an effective regularization tactic to improve the generalization performance of deep learning models.

1.3 The mechanism of KD regularization

In distillation learning, we distill a high-capacity teacher model to obtain a low-capacity student model, hoping that the performance of student is close to that of the teacher. With a capacity equal to that of the teacher, a student model may even

be better than the teacher model[27]; then the student may guide the learning of the teacher model[28]. In some cases, student models can still benefit from the teacher model even if the latter performs poorly[17]. Aside from the above empirically-based results, there also exist theoretical works explaining why knowledge transfer from the ineffective teacher model leads to more powerful student models[29−30]. Soft targets provide regularization via label smoothing training[29,31−32]. In effect, label smoothing regularization (LSR) avoids overfitting by exploring the relationship between ground truth labels and other labels. Thus, even if the teacher does not perform as well as the student, it is still possible to improve the student model by avoiding the overconfidence of the teacher. The student network trained by Furlanello et al.[27] in a KD manner has a capacity equal to that of the teacher network, but performs better than the teacher model; the reason is that soft targets provide regularization by importance sampling weighting, which could very much benefit from correct prediction of the sample confidence by the teacher model.

In the following two sections, we present regularization strategies: Section 2 is on the forward distillation regularization; Section 3 is on mutual distillation regularization that involves peer-to-peer guidance; See Table 2 for a comparison of two strategies.

Table 2 Application scenarios and typical tasks for KD regularization strategies

	Teacher status	Student status	Application scenarios	Typical tasks
Forward distillation	Trained	Untrained	Supervised learning Few shot learning	Image lassification Face recognition
Mutual distillation	–	Untrained	Multi-network learning Few-shot learning Transfer learning	Image lassification

2 Forward Distillation

Forward distillation is the transmission of knowledge from a wider, deeper network with teachers to a narrower, shallower network with students. The training process contains two stages: The teacher network is pre-trained before distillation, and then students draw knowledge from the teacher. The learning process is illustrated in Fig 2 and different types of knowledge transfer are shown in Table 3.

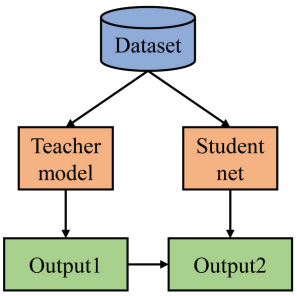

Fig 2 Forward distillation. The teacher transfer knowledge to students[33]

Table 3 Different forms of knowledge transfer in forward distillation

Type of knowledge	References	Brief description
Knowledge of output layer	[15, 18, 34]	Students learn the soft labels of the teacher
Knowledge of middle layer	[35]	Students learn the distribution of the teacher
Structural knowledge	[36]	Students learn structural information from the output of the teacher

Hinton et al.[15] were successful in transferring the knowledge of the teacher to the students. Ahn et al.[35] increased the mutual information (MI) of the middle layer characteristics of the teacher and the student. A higher MI indicates a stronger capability to transfer knowledge. Specifically, given input sample x from the target data distribution $p(x)$ and K pairs of layers $\{(\mathcal{T}^{(k)},\mathcal{S}^{(k)})\}_{k=1}^{K}$, where each set layer $(\mathcal{T}^{(k)},\mathcal{S}^{(k)})$ is drawn from the teacher and the student, respectively. The sample x is passed through the teacher and the student, producing K sets of predictions $\{(\mathbf{t}^{(k)},\mathbf{s}^{(k)})\}_{k=1}^{K}$. The MI between the teacher and the student is defined as

$$I(\mathbf{t};\mathbf{s}) = H(\mathbf{t}) - H(\mathbf{t}|\mathbf{s}) = -\mathbb{E}_{\mathbf{t}}[\log p(\mathbf{t})] + \mathbb{E}_{\mathbf{t},\mathbf{s}}[\log p(\mathbf{t}|\mathbf{s})] \tag{5}$$

where $H(\mathbf{t})$ denotes the entropy value of the teacher, $H(\mathbf{t}|\mathbf{s})$ denotes that of the teacher conditional on the known student, and $\mathbb{E}$ is the expectation. To increase the MI of the output features between the teacher and the student, we minimize the loss function

$$\mathcal{L}_{\text{Total}} = \mathcal{L}_{\text{task}} - \sum_{k=1}^{K} \lambda_k I(\mathbf{t}^k, \mathbf{s}^k) \tag{6}$$

where $\mathcal{L}_{\text{task}}$ is the loss of given tasks and λ_k is a hyperparameter. The student picks up the loss of their specific task while keeping a high MI with the teacher, maximizing MI to stimulate knowledge transfer by learning and estimating the distribution in the teacher.

Park et al.[36] proposed the relational knowledge distillation (RKD), which involves transferring the structured relationships between the output predictions of multiple teacher models into the student model. In contrast to the traditional distillation learning method, which focuses only on the individual the output of the network, combining the outputs of multiple networks into structural units can better reflect the structural properties of the teacher and thus provides more efficient guidance for the student; see Fig 3 and Fig 4. The RKD loss function takes the form

$$\mathcal{L}_{\text{RKD}} = \sum_{(x_1,\cdots,x_n)\in\mathcal{X}^n} \ell(\psi(t_1,\cdots,t_n), \psi(s_1,\cdots,s_n)) \tag{7}$$

where $\mathcal{X}^n$ is a set with n different samples. For a given sample x_i, the output of the teacher and the student are denoted by t_i and s_i, respectively. ψ denotes a relational potential function utilized to extract structural information between the individual model outputs. ℓ is a general loss that measures the difference between the output of the teacher and that of the student. Firstly, the distance relationship distillation loss $\mathcal{L}_{\text{RKD-D}}$ depends on the difference in the distances between the two samples of the teacher and the student.

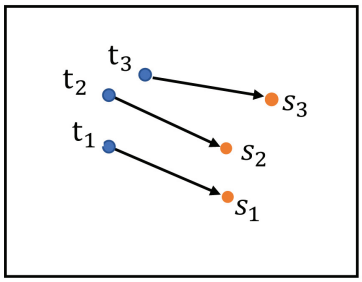

Fig 3 Traditional KD is a point-to-point transfer of knowledge[36]

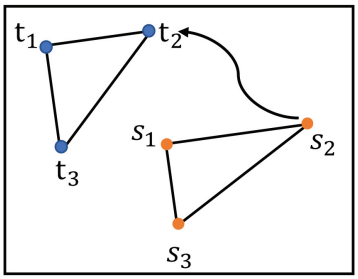

Fig 4 Structural KD is the transfer of knowledge from category to category[36]

$$\begin{cases} \mathcal{L}_{\text{RKD-D}} = \sum\limits_{(x_i,x_j)\in\mathcal{X}^2} \ell_\delta(\psi_D(t_i,t_j), \psi_D(s_i,s_j)) \\ \psi_D(t_i,t_j) = \frac{1}{\mu}\|t_i - t_j\|_2 \\ \mu = \frac{1}{|\mathcal{X}^2|} \sum\limits_{(x_i,x_j)\in\mathcal{X}^2} \|t_i - t_j\|_2 \end{cases} \tag{8}$$

where ψ_D is the Euclidean distance between outputs of models, μ is a parameter for distance normalization, and $\ell_\delta(x,y)$ is the Huber loss,

$$\ell_\delta(x,y) = \begin{cases} \frac{1}{2}(x-y)^2 & , \quad \text{if } |x-y| \le 1 \\ |x-y| - \frac{1}{2} & , \quad \text{otherwise} \end{cases} \tag{9}$$

Secondly, angle-wise distillation loss $\mathcal{L}_{\text{RKD-A}}$ measures three sample angular relationships, which are utilized to transfer the teacher to the student by angular difference corresponding to the feature maps of the training samples:

$$\begin{cases} \mathcal{L}_{\text{RKD-A}} = \sum\limits_{(x_i,x_j,x_k)\in\mathcal{X}^3} \ell_\delta(\psi_A(t_i,t_j,t_k), \psi_A(s_i,s_j,s_k)) \\ \psi_A(t_i,t_j,t_k) = \cos\angle t_i t_j t_k = \langle \mathbf{e}^{ij}, \mathbf{e}^{kj} \rangle \\ \mathbf{e}^{ij} = \frac{t_i - t_j}{\|t_i - t_j\|_2} \quad \mathbf{e}^{kj} = \frac{t_k - t_j}{\|t_k - t_j\|_2} \end{cases} \tag{10}$$

The overall optimization objective function is then

$$\mathcal{L}_{\text{Total}} = \mathcal{L}_{\text{task}} + \lambda_{\text{KD}} \cdot (\mathcal{L}_{\text{RKD-D}} + \mathcal{L}_{\text{RKD-A}}) \tag{11}$$

where $\mathcal{L}_{\text{task}}$ is the loss for given tasks and λ_{KD} is a hyperparameter.

Lukman et al.[34] proposed full deep distillation mutual learning (FDDML) and half deep distillation mutual learning (HDDML). Both methods combine mutual learning and traditional KD to further improve the performance of student networks. In FDDML, the training of each of the two student networks Θ_{s1} and Θ_{s2} benefits not only from the knowledge of the teacher but also from the guidance of the other student. In HDDML, while Θ_{s2} is trained in the same way as that in FDDML, the student Θ_{s1} is trained only from the guidance of another student. Formally, given N training samples $X=\{x_i\}_{i=1}^N$ from M classes, the corresponding label $Y=\{y_i\}_{i=1}^N$ with $y_i\in\{1,2,3,\cdots,M\}$, the total loss functions of FDDML take the form,

$$\begin{aligned}\mathcal{L}_{\text{Total}}^{s1}&=\mathcal{L}_{\text{CE}}^{s1}(\Theta_{s1},X,Y)+\lambda\cdot\mathcal{L}_{\text{distill}}(\mathcal{P}_t,\mathcal{P}_{s1})+\beta\cdot\mathcal{L}_{\text{mimicry}}(p_2\|p_1)\\\mathcal{L}_{\text{Total}}^{s2}&=\mathcal{L}_{\text{CE}}^{s2}(\Theta_{s2},X,Y)+\lambda\cdot\mathcal{L}_{\text{distill}}(\mathcal{P}_t,\mathcal{P}_{s2})+\beta\cdot\mathcal{L}_{\text{mimicry}}(p_2\|p_1)\end{aligned}\tag{12}$$

where λ and β are balance factors, the cross-entropy (CE) loss $\mathcal{L}_{\text{CE}}^{s1}$ and $\mathcal{L}_{\text{CE}}^{s2}$ are defined as

$$\begin{aligned}\mathcal{L}_{\text{CE}}^{s1}(\Theta_{s1},X,Y)&=-\sum_{i=1}^{N}\sum_{m=1}^{M}I(y_i,m)\log(p_1^m(x_i))\\\mathcal{L}_{\text{CE}}^{s2}(\Theta_{s2},X,Y)&=-\sum_{i=1}^{N}\sum_{m=1}^{M}I(y_i,m)\log(p_2^m(x_i))\end{aligned}\tag{13}$$

where $I(y_i,m)=\begin{cases}1, & \text{if } y_i=m\\0, & \text{if } y_i\neq m\end{cases}$ is an indicator function, $p^m(x_i)=\exp(z_i^m/T)/\sum_m\exp(z_i^m/T)$ is the class probability, z_i is a logit, and T is the temperature. The distillation loss function is

$$\mathcal{L}_{\text{distill}}(\mathcal{P}_t,\mathcal{P}_s)=-\frac{1}{N}\sum_{n=1}^{N}\sum_{m=1}^{M}p_t^m(x_i)\log p_s^m(x_i)\tag{14}$$

The mimicry loss $\mathcal{L}_{\text{mimicry}}$ is defined as

$$\mathcal{L}_{\text{mimicry}}(p_2\|p_1)=\sum_{n=1}^{N}\sum_{m=1}^{M}p_2^m(x_i)\log\frac{p_2^m(x_i)}{p_1^m(x_i)}\tag{15}$$

The total loss functions of HDDML take the form,

$$\begin{aligned}\mathcal{L}_{\text{Total}}^{s1}&=\mathcal{L}_{\text{CE}}^{s1}(\Theta_1,X,Y)+\lambda\cdot\mathcal{L}_{\text{mimicry}}(p_2\|p_1)\\\mathcal{L}_{\text{Total}}^{s2}&=\mathcal{L}_{\text{CE}}^{s2}(\Theta_2,X,Y)+\lambda\cdot\mathcal{L}_{\text{distill}}(\mathcal{P}_t,\mathcal{P}_{s2})+\beta\cdot\mathcal{L}_{\text{mimicry}}(p_2\|p_1)\end{aligned}\tag{16}$$

where λ and β are two balance factors.

It is widely acknowledged that in forward distillation the teacher can be more effectively generalized than the student. Knowledge comes from the teacher and is applied to the student; This is a one-way transfer in that the teacher is the exporter of knowledge and the student is the receiver. The training of the teacher is not only time-consuming but can also largely affect the learning outcomes of the student. In addition, it is difficult for the student to fully absorb the incoming knowledge, especially when the capacity gap between the teacher and the student is large.

3 Mutual Distillation

For the scenario in Fig 5 where a large-capacity teacher model is unavailable, Zhang et al.[37] proposed the idea of mutual distillation, where student models improve each other by learning together from some common datasets.

This deep mutual learning (DML) model consists of two student networks Θ_1 and Θ_2. Given N training samples $X=\{x_i\}_{i=1}^N$ from M classes and corresponding label $Y=\{y_i\}_{i=1}^N$ with $y_i\in\{1,2,3,\cdots,M\}$, the probability of a sample x_i in networks Θ_1 and Θ_2 belonging to category m is

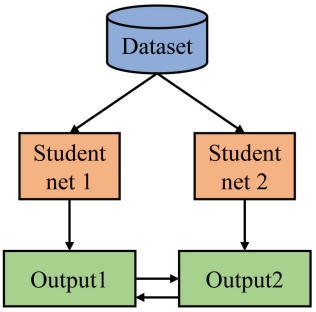

Fig 5 Mutual distillation. Students gain knowledge by learning from each other[33]

$$p_1^m(x_i)=\frac{\exp(z_1^m)}{\sum_{j=1}^{M}\exp(z_1^j)},\quad p_2^m(x_i)=\frac{\exp(z_2^m)}{\sum_{j=1}^{M}\exp(z_2^j)} \tag{17}$$

where z^m denotes the logit output of the m-th category. During the training process, learning experiences are continuously shared between the two networks to achieve simultaneous progress. The total loss of networks Θ_1 and Θ_2 take the form

$$\begin{aligned}\mathcal{L}_{\Theta_1}&=\mathcal{L}_{\mathrm{CE}}^{s1}(\Theta_1,X,Y)+\mathcal{L}_{\mathrm{mimicry}}(p_2\|p_1)\\ \mathcal{L}_{\Theta_2}&=\mathcal{L}_{\mathrm{CE}}^{s2}(\Theta_2,X,Y)+\mathcal{L}_{\mathrm{mimicry}}(p_2\|p_1)\end{aligned} \tag{18}$$

where $\mathcal{L}_{\mathrm{CE}}^{s1}$ and $\mathcal{L}_{\mathrm{CE}}^{s2}$ are cross-entropy loss in equation (13) and $\mathcal{L}_{\mathrm{mimicry}}$ is the mimicry loss in equation (15).

The DML performs well when the network is trained in an end-to-end manner, but it does not fully explore the latent knowledge in the hidden layers. Yao et al.[38] proposed dense cross-layer mutual distillation (DCM), where student networks Θ_{s1} and Θ_{s2} are trained together by attaching classifiers to hidden layers of the two student networks and by carrying out dense two-way KD operations between the layers of classifiers. On the one hand, knowledge is transferred from one student to the other at the same stage layers. On the other hand, two-way KD operations at the different stage layers further stimulate knowledge transfer. After being used during training, these attached classifiers are discarded before inference.

Formally, let $X=\{x_i\}_{i=1}^N$ denote N training samples from M classes and $Y=\{y_i\}_{i=1}^N$ be their corresponding labels with $y_i\in\{1,2,\cdots,M\}$. The set $I=\{(s1_k,s2_k)\}_{k=1}^K$ consists of K pairs of identical stage layer indexes of the student networks. DCM simultaneously optimize Θ_{s1} and Θ_{s2}:

$$\begin{aligned}\mathcal{L}_{\Theta_{s1}}&=\mathcal{L}_c(\Theta_{s1},X,Y)+\alpha\cdot\mathcal{L}_{\mathrm{ds}}(\Theta_{s1},X,Y)+\beta\cdot\mathcal{L}_{\mathrm{dcm}_1}(\hat{\mathcal{P}}_{s2},\hat{\mathcal{P}}_{s1})+\gamma\cdot\mathcal{L}_{\mathrm{dcm}_2}(\hat{\mathcal{P}}_{s2},\hat{\mathcal{P}}_{s1})\\ \mathcal{L}_{\Theta_{s2}}&=\mathcal{L}_c(\Theta_{s2},X,Y)+\alpha\cdot\mathcal{L}_{\mathrm{ds}}(\Theta_{s2},X,Y)+\beta\cdot\mathcal{L}_{\mathrm{dcm}_1}(\hat{\mathcal{P}}_{s1},\hat{\mathcal{P}}_{s2})+\gamma\cdot\mathcal{L}_{\mathrm{dcm}_2}(\hat{\mathcal{P}}_{s1},\hat{\mathcal{P}}_{s2})\end{aligned} \tag{19}$$

where α, β, and γ are hyperparamters. $\mathcal{L}_c$ denotes the classification loss, $\mathcal{L}_{\mathrm{ds}}$ is the overall cross-entropy loss generated by adding attached classifiers to the different phase layers of the student, which take the forms

$$\begin{aligned}\mathcal{L}_{\mathrm{ds}}(\Theta_{s1},X,Y)&=\sum_{k=1}^{K}\mathcal{L}_c(\Theta_{s1_k},X,Y)\\ \mathcal{L}_{\mathrm{ds}}(\Theta_{s2},X,Y)&=\sum_{k=1}^{K}\mathcal{L}_c(\Theta_{s2_k},X,Y)\end{aligned} \tag{20}$$

$\mathcal{L}_{\mathrm{dcm}_1}$ and $\mathcal{L}_{\mathrm{dcm}_2}$ denote the total loss of the same stage bidirectional KD operations and different stage bidirectional KD operations, respectively. The specific expressions take the forms

$$\begin{aligned}\mathcal{L}_{\mathrm{dcm}_1}(\hat{\mathcal{P}}_{s2},\hat{\mathcal{P}}_{s1})&=\sum_{k=1}^{K+1}\mathcal{L}_{\mathrm{kd}}(\hat{\mathcal{P}}_{s2_k},\hat{\mathcal{P}}_{s1_k})\\ \mathcal{L}_{\mathrm{dcm}_2}(\hat{\mathcal{P}}_{s2},\hat{\mathcal{P}}_{s1})&=\sum_{\{(i,j)|1\le i,j\le K+1,i\ne j\}}\mathcal{L}_{\mathrm{kd}}(\hat{\mathcal{P}}_{s2_i},\hat{\mathcal{P}}_{s1_j})\end{aligned} \tag{21}$$

Guo et al.[39] proposed a promising KD method via collaborative learning (KDCL), where multi-student networks with different abilities are jointly trained to produce soft targets of good performance. Specifically, the logits generated by each student are integrated into ensemble logits, which are then used by the teacher to impart knowledge to each student in order to improve the generalization performance of DML.

Gao et al.[40] proposed cross-architecture online-distillation, where multiple students are co-trained in a distributed manner and the logits output of multiple students are integrated into a server to generate soft targets, which are then employed to supervise the regularized training.

Mutual distillation regularization strategies achieve better generalization without teacher models. It can be applied to all kinds of homogeneous or heterogeneous networks. However, there are two limitations to this multi-branch design: Firstly, the training process takes up plenty of storage resources and the number of students is limited by the available memory space; Secondly and more importantly, the model cannot account for a wide range of uncertainty and diversity in the solution space because of the small number of branches.

4 Summarizing Table

In this section, we summarize the bulk of our survey through table, which, hopefully, provides a road map for the rapidly growing field of KD regularization. In Table 4, we compare the four KD regularization strategies with respect to their typical scenarios, advantages, and disadvantages. We also test the performance of representative mutual distillation regularization methods on image classification and collect the benchmark results in Table 5.

Table 4 A comparison of forward distillation and mutual distillation strategies

	Advantages	Disadvantages	Typical scenarios
Forward distillation	Simple and universal	Not applicable to multi-tasks and multi-domains	Pre-trained models are available with low security and privacy
Mutual distillation	Performance is improved as the number of networks increases	Expensive to train multiple networks	Pre-trained models are not available multiple networks and multiple tasks

Table 5 For a fair comparison, all experiments were performed on the same dataset are obtained with the identical settings in the python environment

Datasets	Methods	ResNet-18	ResNet-50	Vgg-16	Vgg-19	DenseNet-121
CIFAR-10	Baseline	94.52	94.59	93.52	93.05	94.50
	DML	S_1: 94.83	S_1: **94.94**	S_1: 93.72	S_1: 93.43	S_1: 94.57
		S_2: 94.86	S_2: 94.72	S_2: 93.76	S_2: **93.55**	S_2: 94.51
	KDCL	S_1: **95.25**	S_1: 94.65	S_1: 93.42	S_1: 93.44	S_1: 94.65
		S_2: 95.18	S_2: 94.56	S_2: **93.87**	S_2: 93.41	S_2: **94.86**
CIFAR-100	Baseline	75.91	77.33	70.48	69.92	76.59
	DML	S_1: 76.08	S_1: 77.87	S_1: 72.78	S_1: 70.49	S_1: 76.17
		S_2: 75.83	S_2: 78.09	S_2: 72.81	S_2: 70.48	S_2: 75.82
	KDCL	S_1: **78.97**	S_1: **78.71**	S_1: **76.00**	S_1: **73.47**	S_1: 79.62
		S_2: 78.79	S_2: 78.10	S_2: 75.01	S_2: 73.32	S_2: **79.82**

The CIFAR-10 dataset[41] consists of 32×32 colour images in 10 classes including 50 000 training samples and 10 000 test samples, where each class has 5 000 training datas. The CIFAR-100 dataset[41] is a more challenging recognition task, which has more classes with fewer samples on each class. The training and test sets are also 50 000 and 10 000 colored natural scene images (32×32 pixels each) drawn from 100 classes. The ResNet-18, ResNet-50, Vgg-16, Vgg-19, and DenseNet-121 are trained to form scratch and optimized via stochastic gradient descent (SGD) with a momentum of 0.9, weight decay is set to 5×10^{-4}, and an initial learning rate is 0.1. The learning rate is divided by 10 at the 50th and 100th epoch, batch size is set to 64. We use typical data augmentation techniques: 32×32 random crops and horizontal flip. We set the accuracy (%) for image classification as the evaluation metric to evaluate the model's generalization performance. The best performance is shown in boldface, where S_1 is the student network and S_2 is another.

5 Conclusion and Future Work

We have systematically reviewed KD regularization strategies in the literature for model generalization; our classification of these KD regularization strategies is based on the teacher-student relationship between the models of knowledge transfer. When a high-capacity teacher model is available, the forward distillation enhances student performance by transferring knowledge from the teacher to the students. When supervising teacher models are unavailable, mutual distillation regularization methods achieve performance improvements through mutual guidance between the students. Details of these two strategies are discussed in sections 2, 3 and summarized in section 4.

Despite its notable successes, KD regularization also still has a number of limitations. In traditional KD regularization, an optimal student model can only be obtained from a given teacher model. With pre-trained teacher models, it is currently only possible to distill the teacher that can perform the same tasks. Most of the existing KD regularization methods have been only applied to classification tasks. Finally, there is a lack of theory for the explanation of empirical facts in KD such as the observation that the performance of student models may still be improved via knowledge transfer from a teacher model with poor performance.

Accordingly, we point out some future trends in the field of KD regularization:

(a) Designing more effective KD regularizers when there is a significant difference in the capacity of the teacher and student model.

(b) Combining more KD regularization with structure regularization techniques to achieve better generalization.

(c) Distilling an all-around teacher model who is good at tackling different challenges from a number of teacher models who can perform different tasks.

(d) Deriving tighter generalization bounds for KD methods to explore factors affecting model generalization and to guide the design of new methods.

References:

[1] MINAEE S, BOYKOV Y Y, PORIKLI F, et al. Image segmentation using deep learning: a survey[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021, 44(7): 3523-3542.

[2] OTTER D W, MEDINA J R, KALITA J K. A survey of the usages of deep learning for natural language processing[J]. IEEE Transactions on Neural Networks and Learning Systems, 2020, 32(2): 604-624.

[3] KROGH A, HERTZ J. A simple weight decay can improve generalization[J]. Advances in Neural Information Processing Systems, 1991, 4: 950-957.

[4] NAKAMURA K, HONG B W. Adaptive weight decay for deep neural networks[J]. IEEE Access, 2019, 7: 118857-118865.

[5] SRIVASTAVA N, HINTON G E, KRIZHEVSKY A, et al. Dropout: a simple way to prevent neural networks from overfitting[J]. The Journal of Machine Learning Research, 2014, 15(1): 1929-1958.

[6] SALIMANS T, KINGMA D P. Weight normalization: a simple reparameterization to accelerate training of deep neural networks[J]. Advances in Neural Information Processing Systems, 2016, 29: 1-9.

[7] WU Y X, HE K M. Group normalization[J]. International Journal of Computer Vision, 2020, 128(3): 742-755.

[8] TAKAHASHI R, MATSUBARA T, UEHARA K. Data augmentation using random image cropping and patching for deep CNNs[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2019, 30(9): 2917-2931.

[9] YUN S, HAN D, OH S J, et al. Cutmix: regularization strategy to train strong classifiers with localizable features[C]//2019 IEEE/CVF International Conference on Computer Vision. Seoual: IEEE, 2019.

[10] ZHONG Z, ZHENG L, KANG G, et al. Random erasing data augmentation[C]//Proceedings of the AAAI Conference on Artificial Intelligence. New York: AAAI Press, 2020.

[11] WANG J, BAO W, SUN L, et al. Private model compression via knowledge distillation[C]//Proceedings of the AAAI Conference on Artificial Intelligence. Honolulu: AAAI Press, 2019.

[12] SUN S, CHENG Y, GAN Z, et al. Patient knowledge distillation for BERT model compression[C]//Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing(EMNLP-IJCNLP). Hong Kong: Association for Computational Linguistics, 2019.

[13] NGUYEN-MEIDINE L T, BELAL A, KIRAN M, et al. Unsupervised multi-target domain adaptation through knowledge distillation[C]//2021 IEEE Winter Conference on Applications of Computer Vision(WACV). Waikoloa: IEEE, 2021.

[14] BA L J, CARUANA R. Do deep nets really need to be deep?[J]. Advances in Neural Information Processing Systems, 2014, 27: 2654-2662.

[15] HINTON G, VINYALS O, DEAN J. Distilling the knowledge in a neural network[J]. Computer Science, 2015, 14177: 38-39.

[16] ZHENG Z, PENG X. Self-guidance: improve deep neural network generalization via knowledge distillation[C]//2022 IEEE/CVF Winter Conference on Applications of Computer Vision. Waikoloa: IEEE, 2022.

[17] CHO J H, HARIHARAN B. On the efficacy of knowledge distillation[C]//2019 IEEE/CVF International Conference on Computer Vision(ICCV). Seoul: IEEE, 2019.

[18] NAYAK G K, MOPURI K R, SHAJ V, et al. Zero-shot knowledge distillation in deep networks[C]//The 36th International Conference on Machine Learning. Long Beach: International Machine Learning Society(IMLS), 2019.

[19] ZHANG L, SONG X, GAO A, et al. Be your own teacher: improve the performance of convolutional neural networks via self distillation[C]//2019 IEEE/CVF International Conference on Computer Vision. Seoul: IEEE, 2019.

[20] CHEN D, MEI J P, WANG C, et al. Online knowledge distillation with diverse peers[C]//The 34th AAAI Conference on Artificial Intelligence. New York: AAAI Press, 2020.

[21] YUN S, PARK J, LEE K, et al. Regularizing class-wise predictions via self-knowledge distillation[C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR). Seattle: IEEE, 2020.

[22] MORADI R, BERANGI R, MINAEI B. A survey of regularization strategies for deep models[J]. Artificial Intelligence Review, 2020, 53(6): 3947-3986.

[23] SANYOS C F G, PAPA J P. Avoiding overfitting: a survey on regularization methods for convolutional neural networks[J]. ACM Computing Surveys(CSUR), 2022, 54(10s): 1-25.

[24] TANG A, QUAN P, NIU L F, et al. A survey for sparse regularization based compression methods[J]. Annals of Data Science, 2022, 9(4): 695-722.

[25] TIAN Y J, ZHANG Y Q. A comprehensive survey on regularization strategies in machine learning[J]. Information Fusion, 2022, 80: 146-166.

[26] PAN S J, YANG Q. A survey on transfer learning[J]. IEEE Transactions on Knowledge and Data Engineering, 2009, 22(10): 1345-1359.

[27] FURLANELLO T, LIPTON Z, TSCHANNEN M, et al. Born again neural networks[J]. Proceedings of Machine Learning Research, 2018, 80: 1607-1616.

(下转第 549 页)