Data-Free Knowledge Distillation with Soft Targeted Transfer Set Synthesis

Zi Wang
Department of Electrical Engineering and Computer Science, The University of Tennessee
zwang84@vols.utk.edu

Abstract

Knowledge distillation (KD) has proved to be an effective approach for deep neural network compression, which learns a compact network (student) by transferring the knowledge from a pre-trained, over-parameterized network (teacher). In traditional KD, the transferred knowledge is usually obtained by feeding training samples to the teacher network to obtain the class probabilities. However, the original training dataset is not always available due to storage costs or privacy issues. In this study, we propose a novel data-free KD approach by modeling the intermediate feature space of the teacher with a multivariate normal distribution and leveraging the soft targeted labels generated by the distribution to synthesize pseudo samples as the transfer set. Several student networks trained with these synthesized transfer sets present competitive performance compared to the networks trained with the original training set and other data-free KD approaches.

Introduction

In recent years, deep neural networks (DNNs) have been widely applied to various applications such as object classification and detection (Krizhevsky, Sutskever, and Hinton 2012; Ren et al. 2015; Huang et al. 2017), image synthesis (Goodfellow et al. 2014; Gulrajani et al. 2017; Li, Wang, and Qi 2018), and robotic control (Levine et al. 2018). However, as state-of-the-art performance is usually acquired by leveraging deeper and wider architectures (Simonyan and Zisserman 2014; Szegedy et al. 2015; He et al. 2016; Huang et al. 2017), over-parameterization becomes a critical issue that prohibits DNN's usage on resource-efficient platforms such as mobile phones and drones (Zhang et al. 2018). Experts have been putting great effort into compressing large, cumbersome DNNs with the approaches such as quantization (Gong et al. 2014; Han, Mao, and Dally 2015), channel pruning (Li et al. 2016; Wang et al. 2019a,b; You et al. 2020), and knowledge distillation (KD) (Hinton, Vinyals, and Dean 2015; Liu, Chen, and Liu 2019; Heo et al. 2019; Jin et al. 2020).

Among all these approaches, KD is a popular scheme that trains a smaller model (student) to mimic the softmax outputs of a pre-trained over-parameterized model (teacher) (Hinton, Vinyals, and Dean 2015). With this approach, the performance of the student model can be improved compared to training the model solely with the cross-entropy loss. However, classic KD methods usually rely on the dataset that contains labeled samples to transfer the knowledge from the teacher to the student, which is a strong constraint because the training set is not always available due to the following reasons. (1) Most importantly, the dataset used for training the teacher model is not often publicly shared because of the concerns of conflict of interest, privacy issues, or business competition (Taigman et al. 2014; Wu et al. 2016). (2) SOTA models are usually trained with extremely large datasets. For example, the popular object classification dataset ImageNet (Deng et al. 2009) contains more than one million training samples that needs more than 100 GB of storage space. Spreading such a large dataset frequently across devices or on the Internet is a heavy burden and a waste of resources.

Data-free KD, or zero-shot KD, was first introduced in (Nayak et al. 2019) to deal with the problem that labeled training samples are missing. Starting from some targets (output probabilities) obtained with certain prior knowledge, noise inputs are optimized to minimize the distances between the softmax outputs obtained by feeding these noise inputs to the teacher network and the targets. Finally, the optimized samples are used for training the student network via a standard KD procedure. The key idea for the implementation of data-free KD is to generate informative pseudo samples that can capture the distribution of the original training samples. For example, (Nayak et al. 2019) models the softmax space of the teacher network as a Dirichlet distribution and generates images by optimizing the noise input to mimic the softmax outputs sampled from the distribution. Generative adversarial networks (Goodfellow et al. 2014) are used in (Chen et al. 2019) for derivating training samples. By considering the pre-trained teacher network as a fixed discriminator, the generator aims to generate samples that cause the maximum response on the discriminator. Although a few studies have been proposed (Nayak et al. 2019; Chen et al. 2019; Micaelli and Storkey 2019), KD in the absence of prior training data is still not well studied and there are clear opportunities to improve on the performance of existing approaches.

In this paper, we present a novel data-free KD approach as an enrichment of this new line of research. Specifically, we

propose to model the output of an intermediate layer of the teacher model with a multivariate normal distribution and obtain the soft targets with the outputs sampling from the distribution. Then the soft targets are used for training sample synthesis by optimizing noise inputs via backpropagation. Finally, the optimized samples are used to train the student network via a standard KD procedure. Different from existing works that directly model the softmax space to obtain the targets, we argue that modeling shallower feature spaces, and feeding the generated intermediate feature representations to the following layers to obtain soft targets helps improve the performance. As features transition from general to specific when the data flow to deeper layers (Yosinski et al. 2014), modeling the output distribution of shallower layers can obtain more generalized soft targets compared to modeling the softmax space directly.

We summarize the main contributions of our study as follows.

- We model the feature space of the teacher's intermediate layer with a multivariate normal distribution and optimize pseudo samples towards the targets sampled from the distribution. By doing so, the quality of the synthesized samples is improved, which helps train the student better.
- We model the output distribution of the shallower layer, rather than directly modeling the softmax space for targets sampling, so that more generalized soft targets can be obtained, which helps improve the performance.
- The proposed approach is evaluated with various benchmark network architectures and datasets and exhibits clear improvement over existing works. Specifically, our student networks trained with the proposed approach achieve 99.08% and 93.31% accuracies without using any original training samples by transferring the knowledge from the teacher networks pre-trained on the MNIST and CIFAR-10 datasets.

Related Work

Traditional Knowledge Distillation

The idea of KD was initially proposed by (Buciluǎ, Caruana, and Niculescu-Mizil 2006) and was substantially developed by (Ba and Caruana 2014) in the era of deep learning. It trains a smaller student network by matching the logits (also called log probability values) before the softmax activations obtained from a cumbersome network. (Hinton, Vinyals, and Dean 2015) extended this idea by softening the softmax output with a scaling factor called temperature, which produces knowledge with higher entropy and improves the performance of the student network. By doing so, (Hinton, Vinyals, and Dean 2015) becomes a generalized case of (Ba and Caruana 2014). Recently, a number of variants have been proposed by adding extra regulations/alignments to the vanilla KD approach. For example, FitNets (Romero et al. 2014) uses ℓ_2-norm to map the intermediate feature representations of the student with the pre-trained teacher so that a deeper and thinner student than the teacher can be well trained. Attention maps are calculated from the teacher's intermediate feature representations in (Zagoruyko and Komodakis 2016) as an extra alignment for knowledge transfer. MEAL (Shen, He, and Xue 2019) proposes to distill the knowledge from multiple teachers via adversarial learning.

Few-Shot and Meta-Data Knowledge Distillation

Due to the storage costs of large scale datasets, efficient KD approaches using limited training samples are investigated by several studies. In (Kimura et al. 2018), the authors first trained a reference model via KD with only a limited amount of training samples. Then a data augmentation approach is proposed to help increase the performance of the student model by generating pseudo samples via the inducing point method (Snelson and Ghahramani 2006). (Ahn et al. 2019) proposes variational information distillation (VID), which aims to maximize the mutual information between the teacher and the student models for few-shot KD. 1×1 convolution layers are added at the end of all the layer blocks of the student model in (Li et al. 2020). By matching the block-level outputs of the teacher and the student models, distilling the knowledge to the student model can be achieved with only a few label-free samples.

Instead of transferring knowledge with labeled samples, alternatives such as meta-data are also used for generating pseudo samples to train the student model. (Lopes, Fenu, and Starner 2017) stores the activation records of certain layers when training the teacher model as the meta-data and uses them to reconstruct the original training samples by optimizing noise inputs, which are then used as the transfer set for training the student. However, releasing such kinds of meta-data along with the pre-trained network is an unusual scenario for most of the applications.

As mentioned before, all the above KD approaches rely on either the labeled data or their surrogates to produce the class probabilities as the matching targets, which are not always available in practice.

Data-Free Knowledge Distillation

Data-free KD, or zero-shot KD (ZSKD), was first introduced in (Nayak et al. 2019), which transfers the knowledge from the teacher to the student without any type of prior information of the training set. ZSKD (Nayak et al. 2019) models the class probabilities with a Dirichlet distribution and generate labels from the distribution to obtain pseudo training samples. Data-Free Learning (DAFL) (Chen et al. 2019) considers the teacher model as a fixed discriminator and trains a generator to generate images that can produce similar softmax outputs from the teacher and the student model. The student model is trained simultaneously with the generator via KD. Adversarial Belief Matching (ABM) was proposed in (Micaelli and Storkey 2019), which trains a generative adversarial network (Goodfellow et al. 2014) to search for samples on which the student model poorly matches the teacher, and then train the student with the generated samples. DeepInversion (Yin et al. 2020) takes the information stored in the batch normalization layers of the teacher to synthesize images that are used as the transfer set.

Our proposed approach is related to ZSKD, but differs from it in the following ways. (1) ZSKD generates soft tar-

geted labels for each specific class by modeling the softmax space with a Dirichlet distribution. Due to the inherent property of Dirichlet distribution, labels that are mismatched with their real categories are produced (see the ablation study and analysis section for details). We consider all the classes as a whole and use a multivariate normal distribution to model the feature space to resolve this problem. (2) ZSKD models the distribution of the softmax space, however, we argue that modeling the data in the shallower feature space generalizes the feature representation better and can therefore improve the performance. (3) We use an extra activation loss term to encourage higher activation values during the image synthesis procedure, which is not used in ZSKD.

The Proposed Approach

In this section, we first briefly present the procedure of standard KD. Then we introduce our proposed data-free KD approach from the following aspects. (1) Generating soft targeted labels by modeling the intermediate feature space with a multivariate normal distribution. (2) Generating pseudo training samples by optimizing the noise inputs towards the generated soft targeted labels with the pre-trained, fixed teacher model. (3) Using the generated samples as the transfer set to train the student model via standard KD.

Knowledge Distillation

Knowledge distillation (Hinton, Vinyals, and Dean 2015) is a popular model compression approach by training a compact student model (S) to mimic the softmax outputs of a pre-trained cumbersome teacher model (T). Let W_T and W_S be the parameters of the teacher and the student model, respectively. The softmax outputs (class probabilities) of the teacher and the student model are represented as $P_T = T(x, W_T) = \text{softmax}(a_T)$ and $P_S = T(x, W_S) = \text{softmax}(a_S)$, respectively, where x is the training sample and a is the pre-softmax activation of a model. During the KD process, a temperature τ is usually used for softening the class probability (Eq. (1)). A larger τ can produce softer class probabilities so that information with higher entropy is enclosed in the targets, and the student model can be trained with a larger learning rate and converge faster.

$$\begin{aligned} P_T^\tau &= T(x, W_T, \tau) = \text{softmax}(\frac{a_T}{\tau}), \\ P_S^\tau &= T(x, W_S, \tau) = \text{softmax}(\frac{a_S}{\tau}). \end{aligned} \tag{1}$$

W_S can be learned by minimizing the loss function in Eq. (2).

$$\mathcal{L}_{KD} = \mathcal{L}_{CE}(P_T^\tau, P_S^\tau) + \lambda_c \mathcal{L}_{CE}(P_S, y), \tag{2}$$

where y is the one-hot ground truth vector, $\mathcal{L}_{CE}(\cdot)$ is the cross-entropy loss, and λ_c is a scaling factor that balances the importance of the two losses.

Data-Free Knowledge Distillation with Soft Targeted Transfer Set Synthesis

In the absence of the original training dataset, pseudo samples have to be generated as the carrier for knowledge transfer. In this study, we propose to model the intermediate feature representation of the teacher model with a multivariate normal distribution. We then use this distribution to generate samples as the feature representations, which are used as either the soft targeted labels, or for producing the soft targeted labels by feeding these samples to the rest layers of the teacher model, depending on whether the softmax space or the intermediate feature space is modeled. We then generate pseudo samples by optimizing the noise inputs through backpropagation with the fixed teacher model. This is achieved by minimizing the distances between the softmax outputs corresponding to the inputs to be optimized and the generated soft targets. Finally, the student model is trained with the pseudo samples through a standard KD process.

Feature space modeling with multivariate normal distribution Suppose the pre-trained teacher model has L layers parameterized with $W_T = \{W_T^1, W_T^2, \cdots, W_T^L\}$. We use a k-dimensional multivariate random variable $\boldsymbol{s^l} = \{s_1^l, s_2^l, \cdots, s_k^l\} \sim p(\boldsymbol{s^l})$ to represent the output space of the l-th layer of the teacher model ($l = 1, 2, \cdots, L$), where s_i^l is the i-th element, and k is the number of elements in the feature space. We model $\boldsymbol{s^l}$ as a multivariate normal distribution, i.e., $\boldsymbol{s^l} \sim \mathcal{N}_k(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu} \in \mathbb{R}^k$ is the mean vector and $\boldsymbol{\Sigma} \in \mathbb{R}^{k \times k}$ is the covariance matrix, respectively.

Statistically, $\boldsymbol{\Sigma} = (\sigma_{ij})_{k \times k}$ can be obtained from its corresponding correlation matrix $\boldsymbol{R} = (\rho_{ij})_{k \times k}$. Let $\boldsymbol{D} = \text{diag}[\sqrt{\sigma_{11}}, \sqrt{\sigma_{22}}, \cdots, \sqrt{\sigma_{kk}}]$, then

$$\Sigma = \boldsymbol{D} \times \boldsymbol{R} \times \boldsymbol{D}, \tag{3}$$

or,

$$\sigma_{ij} = \rho_{ij} \sqrt{\sigma_{ii} \sigma_{jj}}, \quad i, j = 1, 2, \cdots, k. \tag{4}$$

Compared to the covariance matrix $\boldsymbol{\Sigma}$, the coefficient matrix $\boldsymbol{R}$ is a more intuitive statistic that represents the correlation among different components in the random vector, which can be intuitively obtained from the weights of the teacher model. For simplicity, here we consider modeling the feature spaces of the fully connected layers in the teacher model. Suppose we model the feature space of the l-th layer in the teacher model, inspired by (Nayak et al. 2019), we claim that the weights that are used for calculating the l-th layer's feature maps can be considered as a template learned by the teacher model. This template aligns the neurons of the $(l-1)$-th layer to the neurons of the l-th layer and maps the relationship between the feature maps of these two layers. If the value of an element in the feature space peaks, it means that the corresponding weights activate it. On the other hand, if the relationship is misaligned, the value of the certain element decreases. Therefore, we claim that the correlation between the elements of a layer's feature space is implicitly hidden in the weights of this layer, which can be calculated with Eq. (5).

$$R(i, j) = \rho_{ij} = \frac{\boldsymbol{w_i}^T \boldsymbol{w_j}}{||\boldsymbol{w_i}|| \cdot ||\boldsymbol{w_j}||}, \tag{5}$$

where $\boldsymbol{w_i}$ denotes the weights connecting the $(l-1)$-th layer to the i-th element in the l-th layer.

Determining the variance of each variable σ_{ii} $(i = 1, 2, \cdots, k)$ in $\boldsymbol{D}$ is not straightforward in the data-free KD scenario. Therefore, we consider $\boldsymbol{D}$ as a hyperparameter. Actually, σ can be considered as a concentration factor that controls the density of the feature space. A larger σ leads to a distribution that concentrated on one or a few components in the feature representations. On the other hand, when σ gets smaller, the samples are closer to a uniform distribution (see ablation study for details). Moreover, we empirically find that changing the value of $\boldsymbol{\mu}$ only leads to trivial difference of performance. A reasonable explanation on this phenomenon is that $\boldsymbol{\mu}$ only shifts the center of the feature space, which can largely be canceled out by the softmax operation in the last layer. Therefore, in our empirical studies, we just use $\boldsymbol{\mu} = \mathbf{0}$ for simplicity.

Soft targeted label generation Once the feature space of the l-th layer is modeled with a multivariate distribution, and the values of $\boldsymbol{R}$, $\boldsymbol{D}$, and $\boldsymbol{\mu}$ are obtained, we can calculate $\boldsymbol{\Sigma}$ with Eq. (3) and generate samples $\boldsymbol{s^l}$ from the distribution as the l-th layer's outputs of the teacher model. If the last layer (softmax space) is modeled, the soft targeted label $\boldsymbol{y}^{\text{soft}}$ can be obtained with Eq. (6).

$$\boldsymbol{y}^{\text{soft}} = \text{softmax}(\frac{\boldsymbol{s^l}}{\tau}). \tag{6}$$

Otherwise, denote W_T^{l+} the weights after the l-th layer of the teacher model. The soft targeted label can be obtained by feeding $\boldsymbol{s^l}$ to the rest layers of the teacher model (Eq. (7)).

$$\boldsymbol{y}^{\text{soft}} = T(\boldsymbol{s^l}, W_T^{l+}, \tau). \tag{7}$$

We argue that modeling the feature space of a shallower layer can improve the performance, compared to modeling the softmax space (Nayak et al. 2019). Since features gradually transition from general to specific as the tensors feed forward to deeper layers (Yosinski et al. 2014), modeling the feature space of a shallower layer will boost generalization (see ablation study for details).

Sample generation We then use the generated soft labels as the targets to produce pseudo training samples. We first randomly sample n soft targets $\boldsymbol{y_i}^{\text{soft}}$ $(i = 1, 2, \cdots, n)$ and generate a batch of n noise inputs $\hat{x}_i$ $(i = 1, 2, \cdots, n)$. By feeding $\hat{x}_i$ into the fixed teacher model, we obtain the corresponding labels $\hat{\boldsymbol{y}}_{\boldsymbol{i}}$. The noise inputs are optimized with an iterative backpropagation process to minimize the distance between $\boldsymbol{y_i}^{\text{soft}}$ and $\hat{\boldsymbol{y}}_{\boldsymbol{i}}$. There are various criteria that can be utilized to minimize this distance and we choose to use the Kullback–Leibler (KL) divergence (Eq. (8)).

$$\mathcal{L}_d = \mathcal{L}_{KL}(\boldsymbol{y_i}^{\text{soft}}, \hat{\boldsymbol{y}}_{\boldsymbol{i}}), \tag{8}$$

where $\mathcal{L}_{KL}(\cdot)$ is the KL divergence, and $\hat{\boldsymbol{y}}_{\boldsymbol{i}} = T(\hat{x}_i, W_T, \tau)$.

It has been proved that a well trained deep neural network usually receives higher activation values when the training samples are fed. Therefore, we define an extra activation loss $\mathcal{L}_a$ to encourage higher activation values of the last convolutional layer $x_{\text{conv-1}}$ during the image synthesis procedure (Eq. (9)).

$$\mathcal{L}_a = -\frac{1}{n}\Sigma_{i=1}^{n}||x_{\text{conv-1}}^{i}||_1. \tag{9}$$

Algorithm 1 Data-free knowledge distillation for compact student model training

Input: The teacher T with L layers parameterized with W_T, the index of the layer l whose output space is modeled, the temperature τ, the activation loss scaling factor λ_a, the mean $\boldsymbol{\mu}$ and variance $\boldsymbol{D}$ of the elements in the modeled feature space, the number of training iteration N, the batch size n.
Output: The learned student model S.

```
Initialize the transfer set X̂ = ∅
Compute the coefficient matrix R with Eq. (5)
Compute the covariance matrix Σ with Eq. (3)
for i in range(N) do
    Generate n samples: s_n^l ~ N(μ, Σ)
    if l == L then
        y_n^soft = softmax(s_n^l / τ)
    else
        y_n^soft = T(s_n^l, W_T^{l+}, τ)
    Randomly initialize x̂_n and optimize it with Eq. (10)
    X̂ ← X̂ ∪ x̂_n
Using X̂ to train the student model via KD with Eq. (11)
return S
```

Therefore, the total loss of the sample generation process is defined as Eq. (10).

$$\mathcal{L}_{sg} = \mathcal{L}_d + \lambda_a \mathcal{L}_a, \tag{10}$$

where λ_a is a scaling factor that balances the importance of the two terms.

Knowledge distillation with pseudo samples Finally, we use the generated pseudo samples as the transfer set to train the student model with Eq. (11).

$$\mathcal{L}_{DFKD} = \mathcal{L}_{CE}(T(\hat{x}, W_T, \tau), S(\hat{x}, W_S, \tau)). \tag{11}$$

It is worth mentioning that here we omit the second term compared to Eq. (2) when using the transfer set to train the student model. This is because pseudo samples, rather than real training samples, are used for the KD process. Adding a cross-entropy loss with one-hot labels produces little extra meaningful information in this scenario.

We summarize our proposed approach in Algorithm 1.

Experiments

Setup

We evaluate our proposed data-free KD approach on object classification tasks with the following configurations. (1) A LeNet-5 (LeCun et al. 1998) pre-trained with the MNIST dataset (LeCun et al. 1998) following the settings in (Lopes, Fenu, and Starner 2017; Chen et al. 2019) is used as the teacher network. The student model is a LeNet-5-HALF model which contains half the number of filters in each convolutional layer. (2) An AlexNet (Krizhevsky, Sutskever, and Hinton 2012) pre-trained with CIFAR-10 (Krizhevsky, Hinton et al. 2009) as the teacher model and an AlexNet-HALF taking half convolutional filters per layer as the student model. (3) A ResNet-34 (He et al. 2016) pre-trained

with CIFAR-10 as the teacher model and A ResNet-18 as the student model. Architecture details are presented in the Appendix. For each configuration, we first train the teacher model with the cross-entropy loss using a stochastic gradient descent (SGD) optimizer with a batch size of 512 for 200 epochs. The initial learning rate is 0.1, which is divided by 10 at epoch 50, 100, and 150, respectively.

We choose to model the feature space of the second last fully connected layer (represented as FC_{-2} in the following text) of the teacher model and feed the vectors sampled from the distribution to the last layer to get the soft targeted labels. We also implement experiments that model the softmax space for the purpose of performance comparison. We assume the variances for all the elements in the feature space are the same, i.e., $\sigma = \sigma_{11} = \sigma_{11} = \cdots = \sigma_{kk}$ and implement a hyperparameter search to find the optimal value of σ. For transfer set synthesis, we generate a batch of 100 soft labels and noise inputs each time. We optimize the noise inputs by minimizing the KL-divergence between their corresponding softmax outputs with the generated labels with an Adam optimizer (Kingma and Ba 2014) with a learning rate of 0.001 for 1500 iterations. For the activation loss scaling factor λ_a, we implement a hyperparameter search and report the best performance with $\lambda_a = 0.05, 0.05, 0.1$ for the LeNet-5, AlexNet, and ResNet experiments, respectively. We implement data augmentation on the generated samples, and the transformation includes random rotation, padding and random cropping, scaling, translation, and noise adding. All the student models are trained for 2000 epochs with an Adam optimizer (batch size 512, learning rate 0.001) through a standard KD approach. A temperature (τ) of 20 is used for both of the sample synthesis and student model training across all architectures.

All the experiments are implemented with Tensorflow (Abadi et al. 2016) on an NVIDIA GeForce RTX 2080 Ti GPU and an Intel(R) Core(TM) i7-9700K CPU @ 3.60GHz.

Experiments on LeNet-5 with MNIST

MNIST is a handwritten digits dataset, which contains 60,000 training samples and 10,000 test samples with a resolution of 28x28. In our experiments, the samples are resized to 32x32. The results of LeNet-5 with MNIST is reported in Table 1. Note that training the teacher and the student models with standard cross-entropy loss gives us accuracies of 99.32% and 98.99%, respectively. The accuracy of training the student model with a standard KD approach (Hinton, Vinyals, and Dean 2015) is 99.18%, which can be considered as a performance upper bound for data-free KD approaches. Our proposed approach achieves an accuracy of 99.08%, which is very close to the upper bound and outperforms recent data-dependent (few-shot and meta-data-based) approaches (Kimura et al. 2018; Lopes, Fenu, and Starner 2017) with a clear margin. Compared with other data-free KD approaches, our approach also shows competitive performance, which outperforms the previous state-of-the-art ZSKD by 0.31%. It's worth noting that using noise images sampling from a standard normal distribution without any optimization, the accuracy is only 87.58%, which demonstrates that the transfer set generated with the proposed approach precisely captures the distribution of the original training set.

Model	Data	Accuracy
Teacher (standard training)	✓	99.32%
Student (standard training)	✓	98.99%
Standard KD*	✓	99.18%
(Kimura et al. 2018)	●	86.70%
(Lopes, Fenu, and Starner 2017)	■	92.47%
Noise input	✗	87.58%
DAFL	✗	98.20%
ZSKD	✗	98.77%
Ours	✗	99.08%

Table 1: Result on LeNet-5 with the MNIST dataset. Symbols in the "Data" column: ✓: original training data required, ✗: no data required, ●: limited amount of original training data required (few-shot), ■: meta-data required. * indicates the reported results are based on our own implementation.

Model	Data	Accuracy
Teacher (standard training)	✓	78.56%
Student (standard training)	✓	75.29%
Standard KD*	✓	76.88%
Noise input	✗	36.49%
DAFL*	✗	70.23%
ZSKD	✗	69.56%
Ours	✗	73.91%

Table 2: Result on AlexNet with the CIFAR-10 dataset. Symbols in the "Data" column and * have the same meanings as in Table 1.

Experiments on AlexNet with CIFAR-10

We then evaluate our approach with a more challenging dataset, CIFAR-10 on the AlexNet architecture. The CIFAR-10 dataset consists of 50,000 training samples and 10,000 test samples, which are 32x32 color images of common daily-life objects in 10 classes. Because the vanilla AlexNet (Krizhevsky, Sutskever, and Hinton 2012) was designed for the large scale dataset ImageNet (Deng et al. 2009), which takes 227x227 images. We follow (Nayak et al. 2019) to modify the architecture to fit 32x32 inputs. For the first convolutional layer, a 5x5 kernel with a stride of 1 is used. A batch normalization layer is added after each convolutional layer. The modified network contains three fully connected layers, with 512, 256, and 10 neurons, respectively.

The results of AlexNet with CIFAR-10 are presented in Table 2. It can be observed that our proposed approach that generates soft targeted labels with a multivariate normal distribution achieves an accuracy of 73.91%, which is the best among all data-free KD approaches. Specifically, our approach outperforms DAFL (Chen et al. 2019) and ZSKD (Nayak et al. 2019) by 3.68% and 4.35% on the test accuracy, respectively. Since the underlying distribution of the CIFAR-10 training samples is much more complex than the distribution of MNIST, using noise inputs as the transfer set presents a much worse performance on training the student model (36.49%). These results illustrate the effectiveness of

Model	Data	Accuracy
Teacher (standard training)	✓	95.48%
Student (standard training)	✓	93.76%
Standard KD*	✓	94.40%
Noise input	✗	19.49%
DAFL	✗	92.22%
ZSKD*	✗	91.99%
DeepInversion	✗	93.26%
Ours	✗	93.31%

Table 3: Result on ResNet with the CIFAR-10 dataset. Symbols in the "Data" column and * have the same meanings as in Table 1.

our approach on AlexNet with the CIFAR-10 dataset.

Experiments on ResNet with CIFAR-10

We use ResNets to further evaluate our proposed approach in this sub-section. In this experiment, we use a ResNet-34 pre-trained with CIFAR-10 as the teacher model and a ResNet-18 as the student. Similar to the vanilla AlexNet architecture, ResNet was originally designed for the ImageNet dataset, whose inputs are with the resolution of 224x224. We make the following modifications for the CIFAR-10 dataset: (1) we reduce one convolutional layer (and its following batch normalization layer) before the first residual block, (2) the kernel size and stride of the first convolutional layer are changed to 3 and 1, respectively, (3) the kernel size of the average pooling layer after the last residual block is changed to 4 in order to produce 1x1 feature maps.

The performance comparison of the ResNet experiment is reported in Table 3. Our proposed approach brings the test accuracy quite close to the performance obtained by a standard KD procedure with the original training set (93.31% vs. 94.40%). Again, the proposed approach achieves the best performance compared to other data-free KD approaches. It is worth mentioning that our approach achieves similar performance compared to DeepInversion. DeepInversion leverages the statistics in the batch normalization (BN) layer to generate pseudo samples, which can only deal with the scenario in which the teacher has BN layers. Our approach models the intermediate feature space, which is a more generalized case that works for all kinds of networks. Similar to the previous experiments, it can be observed that all the data-free approaches significantly outperform the performance using noise inputs without any optimization to train the student model, which indicates that recovering the prior distribution of the original training samples plays an essential role to achieve good performance.

Ablation Study and Analysis

Performance with Different σs

We first investigate whether different choices of σ have an impact on the performance, with LeNet-5-HALF on MNIST and AlexNet-HALF on CIFAR-10. To test this, we fix $\lambda_a = 0.05$ and generate the transfer set used for training the student model with $\sigma = 0.5, 1.0, 1.5, 2.0, 2.5, 3.0$, respectively. The results are reported in Fig. 1. It appears that the best test

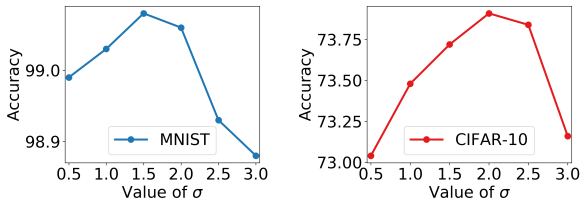

Figure 1: Performance comparison of different choices of σ. Left: LeNet-5-HALF on MNIST. Right: AlexNet-HALF on CIFAR-10.

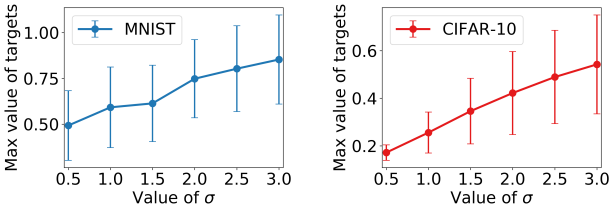

Figure 2: Maximal class probabilities in the soft targets over different σs. Left: LeNet-5-HALF on MNIST. Right: AlexNet-HALF on CIFAR-10. Error bar represents the standard deviation.

accuracy can be obtained when $\sigma = 1.5$ and 2.0 for MNIST and CIFAR-10, respectively. As can be seen from the results, experiments on both configurations exhibit a wide tolerance range of σ choices to achieve good performance. With the σ values from 0.5 to 3.0, all the trained student models present competitive test accuracy compared to previous state-of-the-art (Nayak et al. 2019).

We further investigate the influence of σ selection by plotting the mean maximal class probabilities (i.e, the maximal value in the soft targets) from the soft targets generated with different σs. It is observed that for both configurations, the maximal class probability in the target and its corresponding standard deviation increases as the value of σ grows, which indicates that the targets are more concentrated in one or a few components. A too small σ generates targets in which all the components look similar, which makes them difficult to be distinguished. On the other hand, a too large σ generates targets that are highly concentrated, which leads to less diversity of the soft targets and lower entropy. Both of these two kinds of targets hurt the quality of the generated images. These empirical studies are consistent with the theoretical analysis in the previous section.

Performance with Different λ_a

We then evaluate the effect of different λ_a values. We conduct experiments on LeNet-5-HALF and AlexNet-HALF with $\lambda_a = 0.01, 0.02, 0.05, 0.1, 0.2, 0.5$. The σs are fixed to 1.5 and 2.0, respectively, as used in previous sections to achieve the best performance. Fig. 3 presents the performance of the student models with different values of λ_a. We see that for both configurations, when $\lambda_a = 0.05$, the best test accuracies are achieved, i.e., 99.08% and 73.91% for LeNet-5-HALF and AlexNet-HALF, respectively. For λ_a greater or smaller than 0.05, the test accuracies decrease. When a smaller λ_a is introduced, neurons are usually not

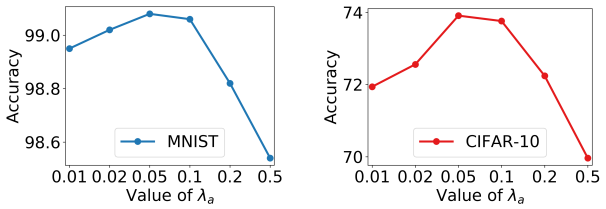

Figure 3: Performance comparison of different choices of λ_a. Left: LeNet-5-HALF on MNIST. Right: AlexNet-HALF on CIFAR-10.

FC_{-2}		✓		✓
Activation			✓	✓
Accuracy	98.92%	99.04%	98.96%	99.08%

Table 4: Performance with different components of the proposed approach on LeNet-5 with the MNIST dataset.

fully activated. On the other hand, for a larger λ_a, the activation values tend to be over-optimized, which in turn prevents the optimization of noise images' softmax outputs from getting close towards the soft targets.

Effect of Each Component in the Loss Function

In this section, we examine the effectiveness of the different components in the proposed approach to the performance of the student model. We evaluate whether modeling the shallower feature space helps improve the image synthesis process by comparing the performance of modeling the feature space of FC_{-2} and the softmax space, respectively. Moreover, we also evaluate whether adding the extra activation loss helps improve the performance of the student model.

Tables 4 and 5 report the performance with different components of the proposed approach on LeNet-5-HALF and AlexNet-HALF, respectively. It is observed that without modeling FC_{-2} and the activation loss, i.e., modeling the softmax space with a multivariate normal distribution, accuracies of 98.92% and 71.95% are achieved with each configuration, respectively. When modeling the output space of the second last full connected layer (FC_{-2}), the performance are improved by 0.12% and 1.59% with LeNet-5-HALF and AlexNet-HALF, respectively. These improvements over their counterparts with ZSKD (98.77% and 69.56%), which models the softmax space with a Dirichlet distribution, validates the effectiveness of modeling the shallower feature space with a multivariate normal distribution. It can be observed that encouraging higher activation values also helps improve the performance, though the improvement is not as significant as that of modeling FC_{-2} instead of the softmax space. With both FC_{-2} and the activation loss implemented, the student models achieve the best performance.

Multivariate Normal vs. Dirichlet Distribution

Since both ZSKD and our approach model the feature space with a prior probability distribution, we further investigate the differences between these two approaches in this subsection. We generate 100 soft targeted labels using each approach with an AlexNet teacher model trained with CIFAR-10 for illustration. Fig. 4 gives the 2-d t-SNE (Maaten and Hinton 2008) embeddings of the soft targeted labels generated by ZSKD (with Dirichlet distribution) and our approach (with multivariate normal distribution). It is observed that samples generated from the multivariate normal distribution are clustered well, which are separable in the low-dimensional space. On the other hand, sampling from the Dirichlet distribution leads to a mixture of targets that belong to different classes in each cluster, which indicates that the generated labels are mismatched with its real category. This is because ZSKD generated labels for each category separately, and there is always a chance that the index corresponding to the maximal probability in the softmax output mismatches the real category. Actually, in our empirical study, Dirichlet distribution can produce around 20% to 40% labels that are mismatched, which substantially hurt the quality of the generated samples. On the other hand, modeling with a multivariate normal distribution considers the samples of all the classes as a whole, which can theoretically avoid the label mismatch problem.

FC_{-2}		✓		✓
Activation			✓	✓
Accuracy	71.95%	73.54%	72.46%	73.91%

Table 5: Performance with different components of the proposed approach on AlexNet with the CIFAR-10 dataset.

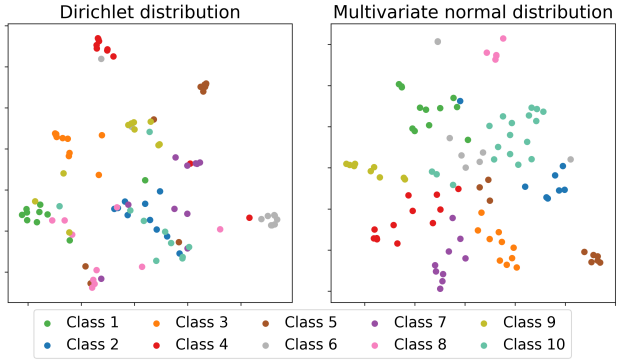

Figure 4: Visualization of the 2-d embeddings of the generated soft targeted labels with ZSKD (Dirichlet distribution) and our proposed approach (multivariate normal distribution) via t-SNE. Figure best viewed in color.

Conclusion

In this paper, we proposed a data-free knowledge distillation approach. We first modeled the intermediate feature space of the teacher model with a multivariate normal distribution and sampling from that distribution to generate soft targeted labels, which are then used to generate pseudo training samples as the transfer set. Finally, the student model is trained with the transfer set via a standard KD process. We evaluate the proposed approach with several benchmark architectures and datasets on the object classification task and the results demonstrate the effectiveness of our approach.

References

Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, 265–283.

Ahn, S.; Hu, S. X.; Damianou, A.; Lawrence, N. D.; and Dai, Z. 2019. Variational information distillation for knowledge transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 9163–9171.

Ba, J.; and Caruana, R. 2014. Do deep nets really need to be deep? In *Advances in neural information processing systems*, 2654–2662.

Buciluǎ, C.; Caruana, R.; and Niculescu-Mizil, A. 2006. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 535–541.

Chen, H.; Wang, Y.; Xu, C.; Yang, Z.; Liu, C.; Shi, B.; Xu, C.; Xu, C.; and Tian, Q. 2019. Data-free learning of student networks. In *Proceedings of the IEEE International Conference on Computer Vision*, 3514–3522.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Gong, Y.; Liu, L.; Yang, M.; and Bourdev, L. 2014. Compressing deep convolutional networks using vector quantization. *arXiv preprint arXiv:1412.6115* .

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, 2672–2680.

Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; and Courville, A. C. 2017. Improved training of wasserstein gans. In *Advances in neural information processing systems*, 5767–5777.

Han, S.; Mao, H.; and Dally, W. J. 2015. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149* .

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Heo, B.; Lee, M.; Yun, S.; and Choi, J. Y. 2019. Knowledge distillation with adversarial samples supporting decision boundary. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 3771–3778.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* .

Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4700–4708.

Jin, X.; Lan, C.; Zeng, W.; and Chen, Z. 2020. Uncertainty-Aware Multi-Shot Knowledge Distillation for Image-Based Object Re-Identification. *arXiv preprint arXiv:2001.05197* .

Kimura, A.; Ghahramani, Z.; Takeuchi, K.; Iwata, T.; and Ueda, N. 2018. Few-shot learning of neural networks from scratch by pseudo example optimization. *arXiv preprint arXiv:1802.03039* .

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images. Technical report, Citeseer.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 1097–1105.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11): 2278–2324.

Levine, S.; Pastor, P.; Krizhevsky, A.; Ibarz, J.; and Quillen, D. 2018. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International Journal of Robotics Research* 37(4-5): 421–436.

Li, C.; Wang, Z.; and Qi, H. 2018. Fast-converging conditional generative adversarial networks for image synthesis. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, 2132–2136. IEEE.

Li, H.; Kadav, A.; Durdanovic, I.; Samet, H.; and Graf, H. P. 2016. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710* .

Li, T.; Li, J.; Liu, Z.; and Zhang, C. 2020. Few sample knowledge distillation for efficient network compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14639–14647.

Liu, J.; Chen, Y.; and Liu, K. 2019. Exploiting the ground-truth: An adversarial imitation based knowledge distillation approach for event detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 6754–6761.

Lopes, R. G.; Fenu, S.; and Starner, T. 2017. Data-free knowledge distillation for deep neural networks. *arXiv preprint arXiv:1710.07535* .

Maaten, L. v. d.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9(Nov): 2579–2605.

Micaelli, P.; and Storkey, A. J. 2019. Zero-shot knowledge transfer via adversarial belief matching. In *Advances in Neural Information Processing Systems*, 9551–9561.

Nayak, G. K.; Mopuri, K. R.; Shaj, V.; Babu, R. V.; and Chakraborty, A. 2019. Zero-shot knowledge distillation in deep networks. *arXiv preprint arXiv:1905.08114* .

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, 91–99.

Romero, A.; Ballas, N.; Kahou, S. E.; Chassang, A.; Gatta, C.; and Bengio, Y. 2014. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550* .

Shen, Z.; He, Z.; and Xue, X. 2019. Meal: Multi-model ensemble via adversarial learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 4886–4893.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* .

Snelson, E.; and Ghahramani, Z. 2006. Sparse Gaussian processes using pseudo-inputs. In *Advances in neural information processing systems*, 1257–1264.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1–9.

Taigman, Y.; Yang, M.; Ranzato, M.; and Wolf, L. 2014. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1701–1708.

Wang, Y.; Zhang, X.; Xie, L.; Zhou, J.; Su, H.; Zhang, B.; and Hu, X. 2019a. Pruning from Scratch. *arXiv preprint arXiv:1909.12579* .

Wang, Z.; Li, C.; Wang, X.; and Wang, D. 2019b. Towards Efficient Convolutional Neural Networks Through Low-Error Filter Saliency Estimation. In *Pacific Rim International Conference on Artificial Intelligence*, 255–267. Springer.

Wu, Y.; Schuster, M.; Chen, Z.; Le, Q. V.; Norouzi, M.; Macherey, W.; Krikun, M.; Cao, Y.; Gao, Q.; Macherey, K.; et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144* .

Yin, H.; Molchanov, P.; Alvarez, J. M.; Li, Z.; Mallya, A.; Hoiem, D.; Jha, N. K.; and Kautz, J. 2020. Dreaming to distill: Data-free knowledge transfer via DeepInversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8715–8724.

Yosinski, J.; Clune, J.; Bengio, Y.; and Lipson, H. 2014. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, 3320–3328.

You, S.; Huang, T.; Yang, M.; Wang, F.; Qian, C.; and Zhang, C. 2020. GreedyNAS: Towards Fast One-Shot NAS with Greedy Supernet. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1999–2008.

Zagoruyko, S.; and Komodakis, N. 2016. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928* .

Zhang, X.; Zhou, X.; Lin, M.; and Sun, J. 2018. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 6848–6856.