Knowledge Distillation with Contrastive Inter-Class Relationship

Chaoyi Yang¹, *, Jijun Zeng¹ and Jinbo Zhang¹

¹Guangdong Power Grid Co., Ltd., Guangzhou, China

*chaoyiyang1806@gmail.com

Abstract. Due to the high computational cost, the application of deep neural networks (DNNs) to the real-time tasks has been limited. A possible solution is to compress the size of the model so that the demand for computation resources can be decreased. A popular method is called knowledge distillation (KD). The basic philosophy behind KD is to transfer the information extracted from the larger teacher network to the smaller student network. The general knowledge transfer strategy is to match the one-to-one logit or intermedia layers between the teacher and student networks, correspondingly. This objective may neglect the informative relationship information between different samples. In this paper, we borrow the idea of metric learning to transfer the contrastive relationship information learned from teacher network to the student. Specifically, we use the well-known Triplet loss to regularize the training of the student network. By the modified negative selection strategy, our Contrastive Knowledge Distillation (CKD) method can efficiently improve the performance of the student network compared with the traditional KD methods. Empirical experiments on KD benchmarks and real-world datasets also demonstrate the superiority of CKD.

1. Introduction

Deep learning (DL) has made impressive achievements in a large range of areas [1-5] due to the very deep and complicated architectures of various deep neural networks (DNNs). Despite their remarkable performance, a significant side effect of the cumbersome model architecture is the extreme demand of the computational resource, which makes it impossible for the deployment of a model on the edge or portable devices, such as mobile phones. Paradoxically, although smaller models require lower computation capacity, the performance of them is usually unsatisfied with some difficult tasks under a complex environment.

To shrink the model complexity while retaining the model performance, model compression has received more and more attention in recent years. The traditional model compression strategy includes parameter pruning [6-8] and model quantization [9-11]. Although the traditional methods can directly alleviate the heavy computation cost to some extent, the architecture of the model is still not modified.

Another type of model compression comes from the idea of transfer learning [12]. As for model compression, knowledge distillation (KD) [13] is one of the most commonly used methods. In the training setting of KD, there are two models, the teacher model, and the student model. By mimicking the so-called soft target (i.e., the output of the teacher model), the performance of the student model can be improved. Since proposed by [13], KD has achieved extensive advances in many learning tasks, such as image classification [14], reinforcement learning [15], and scene recognition [16].
Early KD methods only try to match the lower-level information between teacher and student models. Later works [17, 18] start to focus on the transfer of features extracted from intermedia layers, which may contain much more informative knowledge. [19] further employs an attention map to transfer the information more efficiently. However, the feature maps of DNNs may highly relate to their architectures. DNNs with different architecture designs will vary a lot concerning the feature maps of the same instance due to the different information flow and feature transformation. More importantly, the aforementioned methods only consider the one-to-one instance knowledge transfer. The relationship between different instances is underestimated.

Inspired by the idea of metric learning [20], we propose to transfer additional information about the contrastive relationship between different instances. Intuitively, if the student can learn the relationship information from the teacher, it would be better for the student to capture the global structural knowledge of the training data which is beneficial for downstream tasks. Specifically, our proposed contrastive knowledge distillation (CKD) regularizes the training of the student with a Triplet loss on the representation of data. The Triplet loss enforces the student to resemble the pair-wise contrastive relationship of an instance triple (i.e., anchor instance, positive instance, and negative instance) learned from the teacher model. However, it has been shown that randomly choosing the instance triple is inefficient, thus, followed by previous work, we select the negative instance as the sample which is closest to the positive class within a mini-batch for more evident contrast. In this way, the Triplet regularization can work more efficiently. To verify the effectiveness of CKD, we conduct several experiments on the benchmark KD dataset. The experimental results demonstrate that the proposed CKD can significantly improve the performance of the student model.

2. Preliminary
In this section, we first introduce the basic notations for clarification. Then we present the mathematical formulation of traditional KD methods for better understanding the motivation of our proposed approach.

2.1. Notations
In this paper, we consider the image classification task. Given $n$ training instances $x = \{x_i\}_{i=1}^n \in \mathbb{R}^d$, and their corresponding labels $y = \{y_i\}_{i=1}^n \in \mathbb{R}^c$ (one-hot embedding with $c$ classes), we let $S$ and $T$ represent the student and teacher model, respectively. $f_S(x)$ and $f_T(x)$ are the output of the student and teacher model (i.e., the logits before softmax). Specifically, in this paper, we simply use the cross-entropy (CE) loss as the classification penalty which can be formulated as follows:

$$L_{CE} = -\Sigma_i y_i \log \left( \text{softmax}(f(x)) \right). \quad (1)$$

2.2. Traditional KD methods
As mentioned before, traditional KD methods aim at minimizing the one-to-one difference between the output of $S$ and $T$. Thus, the generic objective of traditional KD can be written as:

$$L_{KD} = \ell \left( f_S(x), f_T(x) \right), \quad (2)$$

where $\ell(\cdot)$ is the loss to assess the difference between $f_S(x)$ and $f_T(x)$. For different purposes, $\ell(\cdot)$ has different forms. For example, the original KD which we denote as OKD [13] uses the Kullback-Leibler (KL) divergence to force the student to follow the output of the teacher:

$$L_{OKD} = KL(\text{softmax}(f_S(x)), \text{softmax}(f_T(x))/\tau), \quad (3)$$

where $KL(\cdot)$ denotes the KL divergence. Note that the OKD method introduces a temperature hyper-parameter $\tau$ to soften the probability distribution. Because, generally speaking, the predicted probability distribution of a well-trained teacher $T$ would be very deterministic to the correct class, while the probabilities of all the other classes would be close to 0. Thus, without the temperature hyper-parameter $\tau$, there may be insufficient information distilled from the teacher, so that the student can be improved very slowly.
Some other works expend the knowledge of the teacher from the logit to the intermedia layers. For example, [18] proposes FitNet to transfer the representations learned from the teacher to the student because the representation generally contains much more information than the logit. Specifically, the regularization of FitNet is defined as:

$$L_{Fr} = \left\| f(x) - g\left(f_S(x)\right) \right\|_2^2,$$

where the slight abuse of notation, $f_S(x)$ and $f_T(x)$ denote the representation extracted by the student and teacher model, respectively. $g(\cdot)$ is a linear transformation function to map the student representation to the same dimension as the teacher representation.

From Eq. (2) and Eq. (3), we can find that the traditional KD methods only focus on the corresponding relation of an individual instance.

3. Contrastive knowledge distillation

To explore the structural contrastive relationship between instances, we propose CKD by using the Triplet loss. Essentially, the Triplet loss can close the gap of instances within the same class, while keeping the instances from different classes far away from each other. Thus, a larger margin between different classes can be achieved.

Concretely, Triplet loss employs a tuple $\{x^a, x^p, x^n\}$ to extract the contrastive information from instances, where $x^a, x^p, x^n$ denote the anchor, positive, negative instance, respectively. The positive pair $(x^a, x^p)$ from the tuple is sampled from the same class, while the negative pair $(x^a, x^n)$ is sampled from different classes. The Triplet loss can be optimized when the difference between the positive pair is small, and the difference between the negative pair is big. We can formalize the standard Triplet loss as follows:

$$L_{Trip} = [D(f(x^a), f(x^p)) - D(f(x^a), f(x^n)) + \alpha]_+,$$

where $D(\cdot, \cdot)$ is a measure of difference, $\alpha$ represents the margin, and $[\cdot]_+$ prevents the loss from being negative. In this paper, we use the angular distance [21] to measure the difference as below:

$$D(f(x^a), f(x^p)) = 1 - \frac{\|f(x^a) - f(x^p)\|}{\|f(x^a)\| \cdot \|f(x^p)\|}.$$

The overall loss function is a combination of the CE loss, traditional KD loss, and the Triplet loss as:

$$L_{CKD} = L_{CE} + L_{KD} + \beta L_{CKD},$$

where $\beta$ denotes the trade-off between the traditional KD loss and the contrastive loss. Fig. 1 illustrates the mechanism of the CKD model.

3.1. Choice of positive and negative pairs

To transfer the contrastive relationship knowledge from the teacher to the student, we have to carefully select the positive and negative pair. Because we want the student to learn the contrastive relationship acquired by the teacher, thus, we utilize $(f_S(x^a), f_T(x^p))$ as the positive pair, and $(f_S(x^a), f_T(x^n))$ as the negative pair. In this way, our proposed CKD can push $f_S(x^a), f_T(x^p)$ close to each other, while keeping $f_S(x^a), f_T(x^n)$ far away, which means that the student model can capture the distance relationship of instances as the teacher model.
Besides, random choosing of negative instances is difficult to converge for the training of the student, because most of the negative instances are easy examples that can make barely contribute to the gradient. Therefore, among all the negative instances in a mini-batch, we choose the nearest one to the anchor.

Figure 2. Illustration of real-world monitoring dashboard data.

4. Experiments
In this section, we empirically verify the performance of the proposed CKD method.

4.1. Experiment settings

4.1.1. Datasets. MNIST [22]: MNIST dataset is composed of 70,000 handwriting digit grey-scale images that have a resolution of 28×28 pixels. We use the original training and testing set partition in our experiment.
- CIFAR10 [23]: CIFAR10 contains 60,000 32×32 RGB images. There are a total of 10 common object classes each of which evenly has 5000 training instances and 1000 testing instances. We adopt the standard widely-used data augmentation scheme as in [2].
- CTW [24]: CTW dataset is a very large Chinese text dataset in the wild. It contains about 1 million Chinese characters from 3,850 unique ones annotated by experts in over 30,000 street view images. We only utilize the CTW dataset to train the model and verify the performance on the characters collected from the computer room monitoring dashboards under a real-world setting.

4.1.2. Implementations. For the MNIST dataset, we only conduct preliminary experiments with very small networks. Specifically, we employ a maxout [25] network which has three convolutional layers (48-48-24 channels) as the teacher network. The student network is twice as deep as the teacher network but with less convolutional channels (6-16-16-16-12-12). The total parameters of the student network are approximately 8% of the teacher network.

For the CIFAR10 dataset, we use the well-known ResNet [2] as the base architecture of the teacher and student networks. Specifically, we use the Resnet14 (Resnet with 14 layers) as the teacher network and Resnet8 as the student network. For the real-world Chinese character recognition experiment, we use Resnet50-Resnet18 as the T-S model pair. For the OKD method, we set the temperature hyper-parameter τ as 4 followings [13]. For the training of the maxout network, we simply follow the implementation in [18]. Note that, different learning rates are applied to the convolutional and fully-connected layers. For the training of Resnet, we utilize the stochastic gradient descent (SGD) optimizer to update the parameters of the network. Specifically, the initial learning rate is set to be 0.05, the momentum and weight decay are set to be 0.9 and $5 \times 10^{-4}$, respectively.

4.2. Main results
We demonstrate the classification accuracy of different KD methods in Table 1. From the table, we can see that the KD-based method can improve the performance of the student network. Besides our
proposed CKD can outperform all the contradistinctive traditional KD methods. Note that in the experiment of MNIST, CKD and FitNet even outperform the teacher model. This is probably because the maxout network is inherently a redundant architecture. Compared with the traditional methods which only transfer the instance-to-instance information, we attribute the superiority of our approach to the contrastive relationship information distilled from the teacher network.

Table 1. Comparison of different KD methods on several datasets.

| Method                  | MNIST (%) | CIFAR10 (%) | CTW (%)  | Dashboard (%) |
|-------------------------|-----------|-------------|----------|---------------|
| Teacher                 | 99.45     | 91.31       | 78.39    | 98.25         |
| Non-KD                  | 98.10     | 87.79       | 76.23    | 92.47         |
| OKD [13]                | 99.35     | 88.35       | 76.94    | 96.33         |
| OKD+CKD (proposed)      | 99.40     | 88.69       | 77.82    | 97.54         |
| FitNet [18]             | 99.49     | 88.89       | 77.24    | 96.58         |
| FitNet+CKD (proposed)   | 99.52     | 89.25       | 77.67    | 97.88         |

4.3. Results under the real-world setting
The aforementioned real-world test data is collected from a computer room monitoring dashboards (see Fig. 2 for example). We deploy the character recognition model on the dashboard to monitor the working state of machines. For faster response and higher accuracy, we use the proposed CKD method to compress the larger teacher model (i.e., ResNet50) to a smaller model (i.e., ResNet18). The student model trained without KD can achieve 76.2% accuracy on the validation set of the CTW dataset when classifying 1001 categories, while trained with the proposed CKD can achieve 77.8% accuracy. On the test real-world collected dataset, more significant accuracy improvement of 5.1% can be observed. The detailed results are shown in the last two columns of Table 1.

5. Conclusion
In this paper, motivated by metric learning, we propose the CKD method which modifies the traditional KD training scheme for model compression. Specifically, we combine the Triplet loss with the traditional KD loss to transfer the contrastive relationship information of different instances. Compared with traditional KD methods, the student model trained by CKD can distill more knowledge from the teacher model, thus better mimic the behavior of the teacher. Experiments on benchmark and real-world datasets empirically demonstrate the superiority of our method. For future work, the effectiveness of more powerful metric learning methods can be investigated.

Acknowledgments
Authors wishing to acknowledge assistance or encouragement from colleagues, special work by technical staff, or financial support from organizations should do so in an unnumbered Acknowledgments section immediately following the last numbered section of the paper.

References
[1] 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 pp 4700-4708
[2] He K, Zhang X, Ren S and Sun J 2016 Deep residual learning for image recognition In Proceedings of the IEEE conference on CVPR 2016 pp 770-778
[3] Krizhevsky A, Sutskever I and Hinton G E 2012 Imagenet classification with deep convolutional neural networks In Proceedings of NIPS 2012 pp 1097-1105
[4] Devlin J, Chang M W, Lee K and Toutanova K 2018 Bert: Pre-training of deep bidirectional transformers for language understanding arXiv preprint arXiv:1810.04805
[5] Mikolov T, Sutskever I, Chen K, Corrado G S and Dean J 2013 Distributed representations of words and phrases and their compositionality In Proceedings of NIPS 2013 pp 3111-3119

[6] 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

[7] Li H, Kadav A, Durandovic I, Samet H and Graf H P 2016 Pruning filters for efficient convnets arXiv preprint arXiv:1608.08710

[8] Liu Z, Li J, Shen Z, Huang G, Yan S and Zhang C 2017 Learning efficient convolutional networks through network slimming In Proceedings of the IEEE ICCV 2017 pp 2736-2744

[9] Courbariaux M, Hubara I, Soudry D, El-Yaniv R and Bengio Y 2016 Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or -1 arXiv preprint arXiv:1602.02830

[10] 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

[11] Wu J, Leng C, Wang Y, Hu Q and Cheng J 2016 Quantized convolutional neural networks for mobile devices In Proceedings of the IEEE Conference on CVPR 2016 pp 4820-4828

[12] Pan S J and Yang Q 2009 A survey on transfer learning IEEE Transactions on knowledge and data engineering, 22(10), 1345-1359

[13] Hinton G, Vinyals O and Dean J 2015 Distilling the knowledge in a neural network arXiv preprint arXiv:1503.02531

[14] Krizhevsky A, Sutskever I and Hinton G E 2012 Imagenet classification with deep convolutional neural networks In Proceedings of NIPS 2012 pp 1097-1105

[15] Schmitt S, Hudson J J, Zidek A, Osindero S, Doersch C, Czarnecki W M and Eslami S M 2018 Kickstarting deep reinforcement learning arXiv preprint arXiv:1803.03835

[16] Zhou G, Fan Y, Cui R, Bian W, Zhu X and Gai K 2018 Rocket launching: A universal and efficient framework for training well-performing light net In Thirty-Second AAAI Conference on Artificial Intelligence

[17] Kulis B 2012 Metric learning: A survey Foundations and trends in machine learning, 5(4), 287-364

[18] Wang J, Zhou F, Wen S, Liu X and Lin Y 2017 Deep metric learning with angular loss In Proceedings of the IEEE ICCV 2017 pp 2593-2601

[19] Krizhevsky A and Hinton G 2009 Learning multiple layers of features from tiny images

[20] Yuan T L, Zhu Z, Xu K, Li C J, Mu T J and Hu S M 2019 A large chinese text dataset in the wild Journal of Computer Science and Technology, 34(3), 509-521

[21] Goodfellow I, Warde-Farley D, Mirza M, Courville A and Bengio Y 2013, February Maxout networks In Proceedings of ICML 2013 pp 1319-1327