Dual bidirectional mutual distillation based on dense connections

Bin Lu
College of electrical and information engineering, Hunan University, Changsha, China
lubin2021@hnu.edu.cn

Abstract. In the field of deep neural network, model compression has been widely used. As a model compression method, knowledge distillation has been studied more and more widely. Most of the existing knowledge distillation methods need a lot of data to train the teacher model in advance, and the interaction between their teacher model and student model is often weak. The existing methods such as DML are constrained by loss to realize soft weight sharing, and there is little work to discuss hard weight sharing. Therefore, a new deep interactive online distillation method is proposed in this paper. In this method, the teacher network generates the auxiliary student network and establishes the connection between the two, and the student network generates the auxiliary teacher network and establishes the connection between the two. Then the two auxiliary networks are combined and trained from scratch. Finally, the performance of the model is improved by using back gradient propagation. The effectiveness of this method is verified by experimental tests on common image classification tasks.

Key Words: Knowledge Distillation, Feature Connections, Dual Learning.

1. Introduction
Deep neural network (DNN) has achieved great success in computer vision processing. Although the performance is attractive, there are a lot of parameters in the popular deep neural network model. This leads to high computational costs and high requirements for device memory. For solving this problem, various network model compression techniques [1] been put forward one after another. Recently, knowledge distillation (KD) is a promising compact and accurate solution series model that has attracted increasing attention. Knowledge distillation is a [2] of model compression techniques for learning compact deep neural network models in which a smaller network (student) is trained to simulate representations of large networks (teachers) with higher accuracy. The popularity of knowledge distillation mainly lies in its simplicity and versatility. Teacher-based student model learning is very simple, and there is no limit to the network structure of both models. By making students' predictions consistent with teachers' predictions, students can improve their performance. Recently, some studies have shown that instead of using pre-trained teachers to train students, it is also effective to learn from each other by peer teaching, which is called online distillation. The traditional distillation approach is a two-stage process that requires pre-training of a strong network of teachers and individual transfers of knowledge to a relatively small and untrained network of students, which requires more training time and computational cost. Online knowledge distillation follows a one-stage end-to-end training strategy.
to optimize the performance of the target network without pre-training high-capacity teachers. On the other hand, in the online mutual distillation, there is no specific teacher-student role, from the beginning of training, all networks learn simultaneously through mutual teaching. The results obtained by the network trained in this online distillation mode are not only better than the separately trained cross-entropy loss network, but also better than the network trained in the traditional offline distillation mode from the pre-trained teacher network.

The existing methods of knowledge distillation are all implemented by mining new information knowledge or designing new distillation loss functions. for this, we propose a new perspective: by designing the connecting path between teachers and students, we train together from scratch and realize the gradient propagation of the displayed layer to layer. The latest progress in deep neural network architecture engineering shows that designing complex feature connection paths, such as residual connections and dense connections across adjacent layers, can achieve better information and better gradient streaming. the training model accuracy and convergence are significantly improved. We speculate that if we can combine the teacher and student networks into a single network at the training stage and can easily separate them after training, this simple principle is crucial to opening the door to the development of a completely new framework for knowledge distillation. To explore this hypothesis, we propose a two-interactive learning distillation approach (DIL). specifically, we divide the DIL into two modules, namely, auxiliary network generation and dense feature distillation. In the first module, we hope that the generated auxiliary network can achieve perfect structural alignment with the given network and have superior capacity. Therefore, we use dynamic additive convolution instead of original convolution. In the second module, we hope to achieve knowledge distillation through layer-to-layer gradient propagation rather than traditional simulation processes. Therefore, we have established a dense feature connection between the student network and the auxiliary teacher network, the teacher network and the auxiliary student network, and after the training, all the connections can be deleted naturally, that is, all the connections only exist in the training stage. Effective experiments show that the proposed DIL achieves consistent and significant accuracy enhancement in various neural networks and datasets. Furthermore, ablation studies and hyper parametric sensitivity studies have been conducted to show DIL effectiveness and stability.

2. Related work
In this section, we summarize the existing methods of knowledge distillation.

2.1. Knowledge Distillation
Knowledge distillation starts with training a powerful teacher model and then encourages students to imitate the softening distribution of the teacher model. In addition to the probability distribution, some other studies try to extract various characteristics of students. KD [3] is also considered as a model compression method, such as pruning and quantization, for training the model efficiency of a small network. Deep mutual learning (DML) [5] proposed a method to train student networks to exchange information with each other by KL divergence loss [4] and to achieve better performance than the original network. In this framework, each student network acts as a teacher network for other student networks. One advantage of this approach is the flexibility to apply any different network architecture. However, this approach can only provide limited information to the target because it does not utilize the rich information from the teacher model during distillation. the On-the-fly native ensemble (ONE) [6] is an online distillation that trains only one multi-branch network while building a powerful teacher model with branch login gating to enhance student network learning. It uses a gated module located on the shared layer and therefore only applies when the branch has the same architecture. Moreover, this distillation method based on logit cannot be well utilized in many visual tasks feature maps [7].
2.2. **Online KD**

Online knowledge distillation: traditional offline methods need to train teacher models in advance, while online methods do not require any pre-trained models. Instead, these networks teach each other by sharing their knowledge throughout the training process. Some of the most recent online distillation methods are DML and ONE, which show satisfactory results [8]. DML simply apply KD losses to each other, and all networks learn from each other in a teacher-student relationship, which can achieve better results than offline distillation. DML disadvantage is the lack of an appropriate teacher role and therefore can only provide limited information to each network. ONE pointed out this drawback of DML, unlike the practice of DML distillation between networks, ONE generate gated integration of training networks and align each network as a target. ONE try to create a powerful [9] teacher logit, that provides a wider range of information. ONE disadvantage is that because it can share the low-level layers of the gating module, it can not train different network architectures at the same time. The existing online distillation methods have a common disadvantage [10] that they all rely only on logit, and ignore the information [11] of feature maps, respectively. Hence, our method makes full use of the feature information [12] of the network by establishing dense feature connections between the networks.

3. **Dual interactive learning**

In this section, we explain the dual interactive learning (DIL) proposed in this paper [13], and introduce its principle and composition in detail. Figure 1 summarizes it.

![Figure 1. Summarizes it.](image)

3.1. **Formulation of DIL**

DIL goal is to achieve knowledge transfer from teachers to students, and the overall optimization goal is

\[
LDIL = LCE(\thetaS, x) + LCE(\thetaT, x).
\]

where LCE (\(\theta_s, x\)) and LCE (\(\theta_t, x\)) are the standard cross-entropy loss functions of the student model and the teacher model, respectively. As shown in the figure 1, DIL achieve the purpose of knowledge transfer by establishing a layer-to-layer dense connection between the student network and the auxiliary teacher network, between the teacher network and the auxiliary student network, and combining the auxiliary teacher network and the auxiliary student network into a network from scratch training. After the training only need to disconnect to get the trained student network.
3.2. Network generation
Different from the existing KD methods, our method does not need to train a good network of teachers in advance. In our method, the auxiliary teacher network is automatically generated by the given student network, the auxiliary student network is automatically generated by the given teacher network, and the auxiliary teacher network is consistent with the given student network at any network depth. The auxiliary student network is consistent with the given teacher network at any network depth.

3.2.1. Auxiliary student generation
In order to ensure that the generated auxiliary student network has superior capacity and perfectly aligned with the given teacher network, all the structural units of the given teacher network are retained when the auxiliary student network is generated. But the original convolution is replaced by dynamic additive convolution. The convolution kernel of a given teacher network is represented as \( K_T = \{W_T^l\}, l \in 1,2,\ldots,L \), where \( L \) is the number of convolution layers. Then the convolution approval generated by the auxiliary student network is expressed as \( K_{as} = \{\sum_{i=1}^{n} \alpha_i^l \vec{w}_i^l\}, l \in 1,2,\ldots,L \), as the \( i \)-th kernel of the \( l \)-th layer, \( \vec{w}_i^l \) has the same shape with \( W_T^l \), \( n \) is the kernel number and \( \alpha_i^l \) is a learnable parameter whose values are automatically learned by Softmax or Sigmoid functions. Because of the characteristics of addition property, the input and output feature dimensions of each convolution layer of teacher-student network are the same by replacing the original convolution with additive convolution.

3.2.2. Auxiliary teacher generation
In order to ensure that the generated auxiliary teacher network has superior capacity and perfectly aligned with the given student network, all the structural units of the given student network are retained when the auxiliary teacher network is generated. But the original convolution is replaced by dynamic additive convolution. The convolution kernel of a given student network is represented as \( K_S = \{W_S^l\}, l \in 1,2,\ldots,L \), where \( L \) is the number of convolution layers. Then the convolution approval generated by the auxiliary teacher network is expressed as \( K_{at} = \{\sum_{i=1}^{n} \beta_i^l \vec{w}_i^l\}, l \in 1,2,\ldots,L \), as the \( i \)-th kernel of the \( l \)-th layer, \( \vec{w}_i^l \) has the same shape with \( W_S^l \), \( n \) is the kernel number and \( \beta_i^l \) is a learnable parameter whose values are automatically learned by Softmax or Sigmoid functions. Because of the characteristics of addition property, the input and output feature dimensions of each convolution layer of teacher-student network are the same by replacing the original convolution with additive convolution.

3.3. Dense feature connection distillation
When the auxiliary student network is generated, it is perfectly aligned with the given teacher network on each layer, and then we establish a layer-to-layer dense connection between the two, expressed as \( \phi_{as} = \phi_{as}^l + \phi_{r,l}^l \), as shown in Figure 1, where \( \phi_{as}^l \) and \( \phi_{r,l}^l \) are the feature maps after the \( l \)-th layer of the auxiliary student network and the teacher network respectively, \( \phi_{at}^l \) is the new feature map of the auxiliary student network after the connection. Similarly, when the auxiliary teacher network is generated, it is perfectly aligned with the given student network on each layer, and we establish a layer-to-layer dense connection between the two, expressed as \( \phi_{at}^l = \phi_{at}^l + \phi_{s,l}^l \), as shown in Figure 1, where \( \phi_{at}^l \) and \( \phi_{s,l}^l \) are the feature maps after the \( l \)-th layer of the auxiliary teacher network and the student network respectively, \( \phi_{at}^l \) is the new feature map of the auxiliary teacher network after the connection. Then, the generated auxiliary student network and the auxiliary teacher network are merged into one network to train together from scratch. The existence and propagation of reverse gradient flow will greatly improve the performance of student network and teacher network. After training, all connections can be removed naturally, and these connections only exist in the training phase.
4. Experiment

In this section, we evaluate our method on the CIFAR-100 and compare its performance with existing knowledge distillation methods. For fair comparison, we used common codes of different KD methods and used the same training and data preprocessing settings throughout the experiment. All experiments were implemented with PyTorch [16]. We added Conv1×1 to each junction path in our DIL because we found that it slightly improved the distillation performance (see the ablation studies section). Based on DIL, we also apply another integrated distillation to the output log (logit) of the two heads of the merged model to further improve the performance of each head, which is called the DIL*.

4.1. Experiments on CIFAR-100

**Dataset.** CIFAR-100[15], containing 50000 training images and 10000 test images of 100 classes are the most popular classification data sets to evaluate the performance of knowledge distillation methods.

**Implementation.** By using a series of classical experimental settings, we did experiments on popular ResNets [14] with different depths and WRNS [16] with different widths, respectively. For the auxiliary teacher model, we use dynamic additive convolution instead of students' standard convolution [17]. For each set-up experiment, we run each method 3 times and report the accuracy of the first 1" average (standard)” method.

**Main results.** In table 1, we provide the average results of the baseline, teachers, and our methods(DIL). Note that the baseline results are slightly higher than those reported in the original paper. For our DIL experiments, we uniformly apply feature connections in the first two stages of ResNets or WRNs (we also provide ablation experiments to analyze where to add connection paths). Regarding the ResNets trunk, we observed DIL increase 1.2%~2.0% absolute accuracy, DIL*1.5%~2.6% increase. This indicates that our design is effective for basic block and bottleneck structures with different depths. Moreover, on the WRN backbone, DIL and DIL* outperformed the baseline by 0.6%~1.1% and 0.7%~1.4%, respectively. Taken together, in fact, DIL significantly improves the performance of each student network while DIL* further provides additional improvements. In most cases, the students trained with our method have better performance than the corresponding teachers. These results verify the effectiveness of our proposed method. WRN-40-2[18] performance is not so powerful because this architecture has broadened network channels.

| Model     | Baseline | Teacher | DIL     | Gain  |
|-----------|----------|---------|---------|-------|
| ResNet20  | 68.78(0.22) | 71.05(0.35) | 70.75(0.29) | 1.97  |
| ResNet32  | 70.80(0.15) | 72.78(0.44) | 72.40(0.14) | 1.6   |
| ResNet44  | 71.88(0.13) | 73.48(0.48) | 73.53(0.16) | 1.65  |
| ResNet56  | 72.29(0.17) | 73.79(0.51) | 73.94(0.08) | 1.65  |
| WRN-40-1  | 71.44(0.14) | 72.48(0.23) | 72.55(0.17) | 1.11  |
| WRN-40-2  | 75.96(0.12) | 77.23(0.19) | 76.55(0.15) | 0.59  |

4.2. Ablation study

In this section, we isolate the effects of each element of our method and compare them with possible variants. All experiments were performed on CIFAR-100 datasets. We used DIL, instead of DIL*, and would not use learning rate preheating in all experiments for better ablation studies. For each set-up experiment, we run the method 3 times and report the accuracy of the first 1" average standard.

**Location of connections.** we explore the effect of increasing the location of dense feature connections. This is important because the semantics and robustness of different locations are different. We consider different settings by adding up to three blocks on the CIFAR100 (Conv2_x, Conv3_x and Conv4_x, expressed as C2, C3 and C4, respectively), respectively. The specific results are detailed in Table 4. We observed that adding connections simultaneously in Conv2_x and Conv3_x can bring the best performance improvement, followed by adding connections only in Conv2_x. These results suggest
that adding connections to shallow layers such as Conv2_x and Conv3_x can migrate information well, while Conv4_x extract higher-level semantics and therefore may have lower robustness, which limits the effect of connection supervision. To some extent, this is consistent with multi-task learning and multi-branch network design, [19] where shallow networks are shared and high-level networks are divided into separate branches.

Comparison with using feature loss. The intermediate feature distance between student and teacher models is commonly used in existing KD methods to represent the feature loss [20]. We do not use feature loss but only dense feature connections. We compare the performance of feature loss and dense feature connections under the same generated teacher network. The results show that using dense connections is superior to using feature loss in our proposed framework.

Impact of the connection transformation. We compare performance when using direct connections and adding Conv1× layer 1 to each connection to improve alignment in feature semantics. We observed that adding Conv1×1 was slightly higher than direct connection, implying that no direct connection of any Conv1×1 can still obtain promising intellectual distillation properties. This is mainly due to our first auxiliary teacher generated module. By using dynamic additive convolution instead of normal convolution, teachers are well aligned with students in network depth, and the input and output feature dimensions of each convolution layer between teachers and students are exactly the same.

Auxiliary teacher structure design. As described in 3.3, auxiliary teachers are automatically generated by replacing convolution kernels in students with linear combinations of several kernels. When knowledge is transferred from the generated teacher to the student, the teacher's ability may affect the performance improvement. From Table 7, we dynamic additive our default settings convolution, abbreviated as DAConv) compared with the other two concern-based alternative SE (Hu et al, 2018) and CBAM [21]. We find that using attention modules such as SE and CBAM to generate teachers can also bring significant improvements to the student network under our proposed framework. By contrast, our default settings achieve higher performance than both options.

5. Conclusion
In this paper, we propose a new method of knowledge distillation: dual interactive learning distillation. It designs and develops the framework of knowledge distillation from a new perspective. The additive convolution comes from the dynamic generation auxiliary network, and the dense feature connection surface is established to improve the network performance through the back propagation of gradient flow. This method does not need to define the loss of distillation and does not need to train the teacher network well in advance. We hope that our work will stimulate future research on knowledge distillation design.

References
[1] J. Moody, S. Hanson, A. Krogh, and J. A. Hertz, “A simple weight decay can improve generalization,” Advances in neural information processing systems, vol. 4, pp. 950–957, 1995.
[2] L. Prechelt, “Automatic early stopping using cross validation: quantifying the criteria,” Neural Networks, vol. 11, no. 4, pp. 761–767, 1998.
[3] T. DeVries and G. W. Taylor, “Improved regularization of convolutional neural networks with cutout,” arXiv preprint arXiv:1708.04552, 2017.
[4] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In Proceedings of the European Conference on Computer Vision (ECCV), pages 181–196, 2018.
[5] Kang G, Li J, Tao D. Shakeout: A New Approach to Regularized Deep Neural Network Training[J]. 2019.
[6] Golnaz Ghiasi, Tsung-Yi Lin, I. Sutskever, and Quoc V. Le, “DropBlock: A regularization method for convolutional networks,” arXiv preprint arXiv:1810.12890, 2018.
[7] Y Tang, Y Wang, Y Xu, et al. Beyond Dropout: Feature Map Distortion to Regularize Deep
Neural Networks[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(4):5964-5971.

[8] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” arXiv preprint arXiv:1207.0580, 2012.

[9] Alex Labach and Hojjat Salehinejad, “Survey of Dropout Methods for Deep Neural Networks,” arXiv preprint arXiv:1904.13310, 2019.

[10] L. Wan, M. Zeiler, S. Zhang, Y. LeCun, and R. Fergus, “Regularization of neural networks using dropconnect,” in Proceedings of the 30th International Conference on Machine Learning. PLMR, 2013.

[11] L. J. Ba and B. Frey, “Adaptive dropout for training deep neural networks,” in Proceedings of the 26th International Conference on Neural Information Processing Systems. NIPS, 2013.

[12] S. Wang and C. Manning, “Fast dropout training,” in Proceedings of the 30th International Conference on Machine Learning. PLMR, 2013.

[13] D. P. Kingma, T. Salimans, and M. Welling, “Variational dropout and the local reparameterization trick,” in Advances in Neural Information Processing Systems 28, 2015, pp. 2575–2583.

[14] He K., Zhang X., Ren S., et al. Identity Mappings in Deep Residual Networks. Springer, Cham, 2016.

[15] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, et al. “Squeeze-and-Excitation Networks,” arXiv preprint arXiv:1709.01507, 2017.

[16] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu and Kaiming He, “Aggregated Residual Transformations for Deep Neural Networks,” arXiv preprint arXiv:1611.05431, 2016.

[17] Andrew G. Howard, Menglong Zhu, Bo Chen, et al. “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications” arXiv preprint arXiv:1704.04861, 2017.

[18] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: In-verted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4510–4520 (2018)

[19] Xiangyu Zhang, Xinyu Zhou and Mengxiao Lin Jian Sun, “ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices,” arXiv preprint arXiv:1707.01083, 2017.

[20] Ningning Ma, A. Krizhevsky, Xiangyu Zhang, Hai-Tao Zheng, Jian Sun, “ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design,” arXiv preprint arXiv:1807.11164, 2018.

[21] Sergey Zagoruyko, Nikos Komodakis, “Wide Residual Networks,” arXiv preprint arXiv:1605.07146, 2016.