A Novel Multi-Knowledge Distillation Approach

Lianqiang LI†, Nonmember, Kangbo SUN†, Student Member, and Jie ZHU†‡, Nonmember

SUMMARY Knowledge distillation approaches can transfer information from a large network (teacher network) to a small network (student network) to compress and accelerate deep neural networks. This paper proposes a novel knowledge distillation approach called multi-knowledge distillation (MKD). MKD consists of two stages. In the first stage, it employs autoencoders to learn compact and precise representations of the feature maps (FM) from the teacher network and the student network, these representations can be treated as the essential of the FM, i.e., EFM. In the second stage, MKD utilizes multiple kinds of knowledge, i.e., the magnitude of individual sample’s EFM and the similarity relationships among several samples’ EFM to enhance the generalization ability of the student network. Compared with previous approaches that employ FM or the handcrafted features leaned from FM, the EFM learned from autoencoders can be transferred more efficiently and reliably. Furthermore, the rich information provided by the multiple kinds of knowledge guarantees the student network to mimic the teacher network as closely as possible. Experimental results also show that MKD is superior to the-state-of-arts.

**key words:** knowledge distillation, deep neural networks, compress, accelerate, autoencoders

1. Introduction

There are a lot of algorithms [1]–[3] for compressing and accelerating deep neural networks. Among them, the knowledge distillation approaches are promising as they can provide extra supervision from the teacher network to the student network. There has been a long line of research on the knowledge distillation approaches. Bucila et al. [4] first proposed to approximate an ensemble of teacher networks with a compact student network. After that, Hinton et al. [5] revisited this idea, and put forward to KD, KD managed to match the softened outputs of the softmax layers between the teacher network and the student network because those of the teacher network can provide more information than raw labels. Further, Romero et al. [6] proposed FitNets, FitNets distilled the FM from the teacher network to improve the performance of the student network. Instead of imitating the FM from the teacher network directly, AT [7] enforced the student network to mimic the teacher network by a manually-designed EFM based on an attention mechanism. Recently, RKD [8] utilized the similarity relationships among several samples’ FM as a new knowledge to enhance the generalization ability of the student network. Although the existing knowledge distillation approaches have gained success to some extent, there are still several shortcomings as follows: (1) By transferring the knowledge from FM directly, the teacher network may distill some useless even harmful information to the student network, let alone the high dimensionality of FM may be a burden to achieve an efficient transfer. (2) The existing approaches for transferring EFM are based on manually-designed mechanisms, which may be not reasonable, and (3) few approach could extract and distill rich information from the teacher network to the student network by several aspects.

To address the above issues, this paper proposes a novel knowledge distillation approach, dubbed multi-knowledge distillation (MKD). MKD consists of two stages. In the first stage, it trains the teacher network with an autoencoder as well as the student network with its corresponding autoencoder. The autoencoders are connected with the intermediate layers of the teacher network and the student network to learn EFM automatically. In contrast to previous approaches that employ FM or the handcrafted features leaned from FM, employing the EFM generated by autoencoders in MKD is more efficient and reliable. In the second stage, MKD fine-tunes the student network with multiple pre-defined knowledge from the EFM. More concretely, MKD not only transfers the magnitude of individual sample’s EFM but also distills the similarity relationships among several samples’ EFM. By learning the multi-knowledge, the student network can mimic the teacher network as closely as possible. To validate the effectiveness of MKD, extensive experiments have been conducted, and the experimental results show that MKD is superior to the state-of-the-arts [5]–[8].

2. The Proposed Approach

In this section, we will introduce the proposed MKD in detail.

2.1 Knowledge Extraction

MKD wants to distill the important information from the teacher network’s FM to the student network. As a result, the first concerning of MKD is how to define the important information of FM. Autoencoders [9] are the natural choice to find the key information automatically. A general autoencoder is consisted of an encoder and a decoder, the encoder and the decoder are trained alternatively to improve each
For knowledge distillation problem, taking an example of the teacher network (the same to the student network). As shown in Fig. 1, we connect an autoencoder with its intermediate layer to learn EFM. The encoder first maps FM to the precise and compact representations EFM. Then the decoder maps the EFM to reconstructed FM. The parameters of the autoencoder will be updated through backpropagation by lessening the reconstruction error between FM and FM.

\[ L_{Autoencoder} = \|FM - FM_r \|^2 \]  

By restricting the dimensionality of the output of encoder to lower than that of the input space, the attached autoencoder can learn a series of low-dimensional features that capture the most important information of FM, i.e., EFM.

In the knowledge extraction stage, there are three issues we need make them notably here. First, how to choose the FM? Although the activations of any intermediate layer can be candidates, we choose the outputs from the last convolutional layer as the FM. The intuition is based on the observation that deep neural networks generate features layer by layer and higher layer features are closer to the useful features for performing the main task\[10\]. Second, how to train the networks? In practice, we train the autoencoder and the teacher network or the student network separately, which means that the cross-entropy error shown in Fig. 1 will only be used to update the autoencoder network while the red boxes mean that the parameters there would not be changed during the knowledge transferring process.

2.2 Knowledge Transferring

After obtaining the EFM, the concerning of MKD becomes how to ensure the knowledge from the teacher network is rich enough to improve the performance of the student network. In order to solve this concerning, we propose two kinds of knowledge for transferring, as shown in Fig. 2.

Suppose there are \(N\) samples \(\{(1,...,i,...,j,...,N) | i \neq j\}\) in a batch \(\psi\). \(E^T_i\) denotes the EFM of sample \(i\) from the teacher network while \(E^S_i\) denotes that from the student network.

2.2.1 Magnitude Knowledge

The first transferred knowledge is straightforward yet effective. We enforce \(E^S_i\) to simulate the magnitude of \(E^T_i\). However, the difference in scales between them are usually significantly different. To solve this problem, we introduce an average normalization. Taking an example of the teacher network, the normalized \(E^T_i\) is shown as follows:

\[ \overline{E^T_i} = \frac{E^T_i}{\sum_{i \in \psi} E^T_i} \]  

Then, we transfer the magnitude knowledge by minimizing Eq. (3):

\[ L_{ML} = \| \overline{E^T_i} - \overline{E^S_i} \|^2 \]  

Compared with conventional approaches, i.e., FitNets [6] and AT [7], the proposed magnitude knowledge has two obvious advantages. On the one hand, the dimension of EFM is typically smaller than that of FM, which would be efficient for transferring. On the other hand, due to the characteristics of autoencoders, there would be less redundant information provided by the EFM of the teacher network.

2.2.2 Relationship Knowledge

In order to train the student network more robust, we also compute the similarities among several samples’ EFM as a relationship knowledge. Taking an example of the teacher network, we explore the similarity relationship between the
EFM of any two samples from batch $\psi$ by using a distance-wise potential, as shown in Eq. (4):

$$R^{(l,j)}_T = ||E^{l}_T - E^{j}_T||^2$$

(4)

Similarly, we carry out the average normalization described in Sect. 2.2.1, the normalized $R^{(l,j)}_T$ is rewritten as follows:

$$\overline{R^{(l,j)}_T} = \frac{R^{(l,j)}_T}{\sum_{i\in\psi,j\in\psi} R^{(l,j)}_T}$$

(5)

We denote that of the student network as $\overline{R^{(l,j)}_S}$. Then, the relationship loss is defined as Eq. (6):

$$L_{RL} = \sum_{i\in\psi,j\in\psi} ||\overline{R^{(l,j)}_T} - \overline{R^{(l,j)}_S}||^2$$

(6)

By minimizing Eq. (6), the proposed relationship knowledge is able to transfer information of high-order properties, such as the structural information among several samples’ EFM, which would be complementary to the magnitude knowledge to provide sufficient guidance to the student network.

At last, the overall loss function for fine-tuning the student network is consisted of a task-specific loss, the proposed magnitude loss and relationship loss, which is shown in Eq. (7):

$$L_{Overall} = L_{task} + \lambda_{ML}L_{ML} + \lambda_{RL}L_{RL}$$

(7)

where $L_{task}$ is the task-specific loss at hand, $\lambda_{ML}$ and $\lambda_{RL}$ are the hyper-parameters to balance the loss terms.

During the knowledge transferring process, we treat the EFM generated by the autoencoder connected with the teacher network as the baseline, and we freeze the parameters of the teacher network and its corresponding autoencoder. Further, as the goal of the proposed method is to promote the performance of the student network, and the autoencoder connected with the student network has been trained well in the knowledge extraction process. Thus, we also freeze the parameters of the autoencoder connected with the student network. Finally, the overall loss is pushed to the student network to enforce the student network to generate the FM which is more similar to that produced by the teacher network to enable the student network to have better performance.

3. Experiments

In this section, we will validate the effectiveness of the proposed MKD. It is worth noting that whether the autoencoders can reconstruct the original FM is the basis for MKD to work and the key that enables MKD to be different from other knowledge distillation approaches. Fortunately, the spatial sizes from the outputs of the last convolutional layers are relatively small, experiments have proven that the reconstruction quality are not sensitive to the structures of the autoencoders if there are at least two layers for the encoders and the decoders, respectively. Therefore, we use the autoencoder shown in Fig. 3 for the all experiments. The all experiments are conducted on PyTorch framework [11] for 5 times, and we take the median results as the final results.

3.1 Comparison Experiments

To prove the generality and effectiveness of the proposed method, we first evaluate the proposed MKD by comparing it with other knowledge distillation approaches [5]–[8] on different datasets, i.e., CIFAR-10 and CIFAR-100 and heterogeneous networks, i.e., VGGNet and ResNet. We apply the random rotation and the horizontal flip for data augmentations. The coefficients in Eq. (7) are set 1 to ensure the student network can learn different kinds of knowledge equally.

For CIFAR-10, ResNet56 with 93.61% accuracy and VGG16 with 92.51% accuracy are used as the teacher networks. For CIFAR-100, ResNet110 with 73.09% accuracy and VGG16 with 72.32% accuracy are used as the teacher networks. For both datasets, ResNet20 and ResNet14 are used as the student networks. All of the networks are trained-from-scratch for a fair comparison.

The accuracy performance of different knowledge distillation approaches are summarized in Table 1. We can notice that MKD can enhance the performance of different student networks both on CIFAR-10 and CIFAR-100 datasets. For example on CIFAR-100, the accuracy of ResNet14 educated by MKD is 65.36%, which is 1.56% higher than that of the ResNet14 trained from scratch. Compared with other approaches, MKD is also competitive. In fact, MKD always achieves the best accuracy. This is because the employed autoencoders enable MKD to understand the essential parts of the FM and the defined multiple kinds of knowledge guarantee the student network can mimic the teacher network through several aspects. By contrast, the improvements brought from KD [5], FitNets [6] and AT [7] are relatively small due to their either inefficient or unreasonable knowledge transferring process. As far as the RKD [8] is concerned, transferring the way that how the teacher network handle several samples rather than individual sample makes its performance next to that of the proposed MKD.

3.2 Ablation Study

Aiming to verify the effectiveness of the magnitude knowledge and the relationship knowledge more clearly, we carry out an ablation study on CIFAR-10 and CIFAR-100. The
experimental settings are same as those in Sect. 3.1.

Table 2 presents how the accuracy varies against the two knowledge. It is clear that the two knowledge could get similarly satisfactory results. For example on CIFAR-10, the magnitude knowledge helps ResNet20 get 0.72% improvement while the relationship knowledge helps ResNet20 get 0.76% improvement compared with the ResNet20 trained from scratch. This means both of the two knowledge have contributions to the improvements of MKD. However, the performance of individual knowledge is worse than that in employing the two knowledge, which shows the importance of simultaneous training.

4. Conclusion

In this paper, we propose a novel knowledge distillation approach, named multi-knowledge distillation (MKD) that employs autoencoders to learn and transfer multiple kinds of knowledge from the teacher network to the student network. Experimental results show that MKD achieves notable improvements than that is trained from scratch, and MKD is superior to other knowledge distillation approaches.

In the future work, we would like to utilize the reinforcement learning to choose the most useful knowledge to further narrow and even eliminate the performance gap between the teacher network and the student network.

Acknowledgments

This work was supported by the National Key Research Project of China under Grant No. 2017YFF0210903, the National Natural Science Foundation of China under Grant Nos. 61371147 and 11433002.

References

[1] R. Zhao, Y. Hu, J. Dotzel, C. De Sa, and Z. Zhang, “Improving neural network quantization without retraining using outlier channel splitting,” International Conference on Machine Learning, pp.7543–7552, 2019.
[2] L. Li, J. Zhu, and M.T. Sun, “A spectral clustering based filter-level pruning method for convolutional neural networks,” IEICE Trans. Inf. & Syst., vol.102, no.12, pp.2624–2627, 2019.
[3] Y. Chen, N. Wang, and Z. Zhang, “Darkrank: Accelerating deep metric learning via cross sample similarities transfer,” Thirty-Second AAAI Conference on Artificial Intelligence, 2018.
[4] C. Bucilă, R. Caruana, and A. Niculescu-Mizil, “Model compression,” Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pp.535–541, ACM, 2006.
[5] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” NIPS Deep Learning and Representation Learning Workshop, 2015.
[6] A. Romero, N. Ballas, S.E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, “Fitnets: Hints for thin deep nets,” ICLR, 2015.
[7] S. Zagoruyko and N. Komodakis, “Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer,” ICLR, 2017.
[8] W. Park, D. Kim, Y. Lu, and M. Cho, “Relational knowledge distillation,” Conference on Computer Vision and Pattern Recognition, pp.3962–3971, 2019.
[9] Y. Bengio, “Learning deep architectures for ai,” Foundations and Trends® in Machine Learning, vol.2, no.1, pp.1–127, 2009.
[10] M.D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” European conference on computer vision, pp.818–833, Springer, 2014.
[11] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, “Automatic differentiation in pytorch,” 2017.