JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 36, 1339-1351 (2020)
DOI: 10.6688/JISE.202011_36(6).0013

Student-Centric Network Learning for Improved Knowledge Transfer

HONG-JI WANG[1], XIANG XU[2], BAO-MIN XU[1], SHUANG-YUAN YU[1] AND QUAN-XIN WANG[3]
[1]*School of Computer and Information Technology*
Beijing Jiaotong University
Beijing, 100044 P.R. China
[2]*School of Computing Science*
Simon Fraser University
Vancouver, V5B3J1 Canada
[3]*Department of Computer Science*
Beijing Jiaotong University Haibin College
Huanghua, 061199 P.R. China
E-mail: {18120466; bmxu;shyyu; qxwang}@bjtu.edu.cn; xuxiangx@sfu.ca

In the context of model compression using the student-teacher paradigm, we propose the idea of student-centric learning, where the student is less constrained by the teacher and able to learn on its own. We believe the student should have more flexibility during training. Towards student-centric learning, we propose two approaches: correlation-based learning and self-guided learning. In correlation-based learning, we propose to guide the student with two types of correlations between activations: the correlation between different channels and the correlation between different spatial locations. In self-guided learning, we propose to give the student network the opportunity to learn by itself in the form of additional self-taught neurons. We empirically validate our approaches on benchmark datasets, producing state-of-the-art results. Notably, our approaches can train a smaller and shallower student network with only 5 layers that outperforms a larger and deeper teacher network with 11 layers by nearly 1% on CIFAR-100.

Keywords: knowledge transfer, student-teacher paradigm, correlation-based learning, self-guided learning, dense convolution

1. INTRODUCTION

While deep neural networks have demonstrated superior performance in various computer vision tasks, model compression techniques are needed to make them suitable for use on everyday devices. Current state-of-the-art approaches for computer vision tasks such as image classification, object detection, and semantic segmentation usually involve large neural networks with tens of millions of parameters. These large networks typically require specialized processing units with significant on-board storage memory. As a result, these large networks are not suitable for deployment on user devices with limited computing power and memory such as cell phones and embedded electronics. Model compression techniques must be developed to retain the performance of these networks while reducing the resources needed to run them.

One line of work on model compression attempts to train a smaller student network based on a larger and more powerful teacher network. The goal of this student-teacher

Received February 17, 2019; revised May 16 & July 16, 2019; accepted October 31, 2019.
Communicated by Jen-Hui Chuang.

paradigm is to guide the training of the student network by transferring the knowledge from the teacher to the student. The student network is also designed to be more compact than the teacher network, making the student network more suitable for deployment on devices with limited computing power.

Early student-teacher paradigm-based approaches [1-3] teach the student network to mimic the output of the teacher directly. Recently, FitNets [4] showed that matching intermediate representations learned by the teacher network is also helpful for training the student network. In the training procedure of FitNets [4], an intermediate student layer is forced to mimic the activations of an intermediate teacher layer through a convolutional regressor layer. Unfortunately, forcing the student to directly match the activations of the teacher is a very strong constraint that can actually hinder the performance of the student. Zagoruyko *et al.* [5, 6] recently show that attention maps (*i.e.*, some statistics computed based on the activations of intermediate teacher layers) are better than direct activation matching for transferring knowledge. Therefore, we would like to develop approaches of knowledge transfer that follows a more student-centric learning paradigm where the student network is less constrained to match the teacher network and is freer to learn on its own.

Towards student-centric learning, we explore two approaches to give the student more flexibility during training.

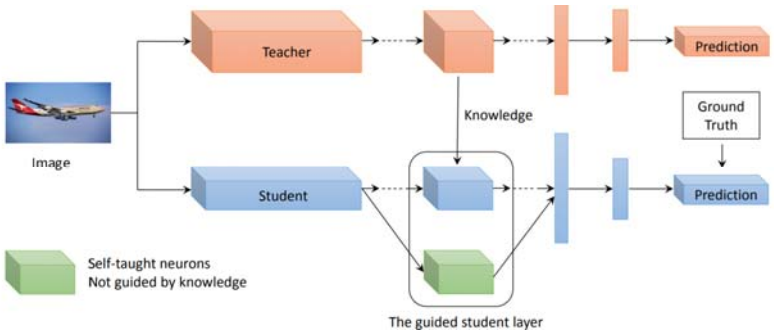


Fig. 1. Self-guided learning. In the guided student layer, self-taught neurons, denoted by the green block in the figure, are trained from scratch with only the supervisory signals from the data. Other neurons in the guided student layer, denoted by the light blue block in the figure, receive knowledge from the teacher.

1.1 Correlation-based Learning

Instead of forcing the student to directly match the activations of the teacher network, we give the student network access to the correlation between the activations of an intermediate teacher network layer. This enables the student network to learn its own intermediate feature representations while retaining similar correlation patterns with the teacher. Particularly, we explore two types of correlations between activations: the correlation between different channels and the correlation between different spatial locations.

1.2 Self-guided Learning

We give the student network the opportunity to learn by itself in the form of additional

self-taught neurons. Procedurally, as shown in Fig. 1, we append the guided student layer with additional self-taught neurons. We call them self-taught neurons because they are trained from scratch using only the supervisory signals from the data (not the teacher). Self-taught neurons allow for the discovery of complementary intermediate representations to the knowledge provided by the teacher, which are beneficial for solving the target task. Notably, self-guided learning is generic and can be combined with correlation-based learning or any other forms of knowledge.

The contribution of our work is summarized as follows. Towards student-centric learning, we propose two approaches: correlation-based learning and self-guided learning. In correlation-based learning, we propose to use the correlation between activations as a new form of knowledge being transferred from the teacher to the student. In self-guided learning, we propose the principled idea of self-taught neurons to allow the student to have more flexibility during training. We empirically validate our approaches on benchmark datasets, achieving state-of-the-art performance. Notably, our approaches can train a smaller and shallower student network with only 5 layers that outperforms a larger and deeper teacher network with 11 layers by nearly 1% on CIFAR-100.

2. RELATED WORK

Model compression using the student-teacher paradigm is pioneered by Bucilua *et al.* [2]. In [2], the authors propose to train a compact neural network to mimic the function learned by a large and complex ensemble. This idea is recently adopted by [1] to train a shallow student network by regressing the logits produced by the teacher network. More recently, in Knowledge Distillation (KD) [3], the authors propose to use the soft target distribution produced by the teacher to guide the student. Hinton *et al.* [3] also show that regressing the logits [1] is a special case of KD.

While the above approaches mainly train the student network to mimic the output of the teacher network, FitNets [4] finds that the intermediate representations learned by the teacher can also help improve the training process and final performance of the student. Particularly, they use the intermediate representations of the teacher network as hints to train the student network such that the student's intermediate representations can match the teacher's intermediate representations through a convolutional regressor layer. Inspired by the critical role of attention mechanism in human visual experience, Zagoruyko *et al.* [5] demonstrate that transferring the attention maps of the teacher network, *i.e.*, where it looks at, is better for training the student network than the activation matching used in FitNets [4]. Different from them [4, 5], we propose to use the correlation between activations as the hints for training the student network, demonstrating that the correlation between activations is better than the direct activation matching used in FitNets and [4] the attention maps used in [5].

Using correlation between activations as a form of knowledge is loosely related to previous work on neural style transfer [7]. They propose to use the feature correlation as the style representation of images, but they only use the correlation between different channels, which discards the spatial structure of images. In addition to the correlation between different channels, we also propose to use the correlation between different spatial locations as hints for training the student network.

Our idea of self-guided learning is related to a recent work on transfer learning [8]. Wang *et al.* [8] find that increasing model capacity during fine-tuning can help existing neurons better adapt and specialize to the target task and significantly improve the performance. Inspired by them, we believe adding self-taught neurons in the student network can allow the student to discover complementary cues to the knowledge provided by the teacher, which are beneficial for solving the target task.

Besides the student-teacher paradigm-based approaches, there are several other approaches proposed towards efficient training and inference of deep neural networks. Representative approaches include pruning redundant weights and connections in an existing network [9-13], quantizing high precision parameters to fixed-point values [14-17], and binarizing the weights and activations in neural networks [18-20]. Our work is related to them but typically the main network architecture remains bulky in these approaches. Compared to them, the student-teacher paradigm-based approaches can produce a much smaller network by transferring the knowledge contained in the large network to the small one. Moreover, the aforementioned approaches can be applied to the small student network obtained under the student-teacher paradigm to further reduce its size.

3. APPROACH

In this section, we detail our proposed approaches for student-centric learning: correlation-based learning and self-guided learning. We first introduce how we guide the student with the correlation between the activations of an intermediate teacher network layer. Then we describe how we give the student the opportunity to learn by itself in the form of additional self-taught neurons.

3.1 Correlation-based Learning

To study the student-teacher paradigm, one must properly specify the form of the knowledge contained in the teacher network. Directly transferring the teacher's activations to the student leaves very few flexibilities for the student to learn the model itself. Since the teacher network is not guaranteed to be optimal, this may hinder the performance of the student network. Therefore, we propose to use the correlation between activations as a form of knowledge. This enables the student network to learn its own intermediate feature representations while retaining similar correlation patterns with the teacher.

Note that we are not simply computing the correlation between each dimension of the activations as this will result in a huge correlation matrix. Matching the huge correlation matrix imposes as strong constraints on the student as transferring the activations and is also computationally expensive. For a layer in the network, the activations can be indexed by their corresponding channel and spatial location. We take advantage of this fact and propose to use two types of correlations as the knowledge: the correlation between different channels and the correlation between different spatial locations.

Formally, the activation of a layer l is a three-dimensional array of size $N_l \times H_l \times W_l$ where N_l is the number of channels or feature maps in layer l, H_l and W_l are spatial dimensions of the feature map. We can reshape the activation array to a matrix $F^l \in \mathbb{R}^{N_l \times H_l}$, where M_l is the number of spatial locations in one feature map and equals to $H_l \times W_l$.

The correlation between channels is given by the Gram matrix $C^l \in \mathbb{R}^{N_l \times N_l}$, where C^l_{ij} is defined as the inner product between the feature activations in channel i and channel j:

$$C^l_{ij} = \sum_{k=1}^{M_l} F^l_{ik} F^l_{jk}. \tag{1}$$

Before transferring the matrix C^l to the student, we first normalize it. For an instance of matrix C^l, we compute the mean and standard deviation based on all the elements in C^l. Then we normalize C^l by subtracting the mean from each element and divide each element by the standard deviation. After that, we transfer the normalized C^l computed based on the activations of a teacher hidden layer to a student hidden layer with L_2 loss. Formally, the L_2 loss function is defined as follows:

$$L_{\text{Channel}} = \| vec(norm(C_S^{l_S})) - vec(norm(C_T^{l_T})) \|_2, \tag{2}$$

where $norm(\cdot)$ refers to the normalization operation described above, $vec(\cdot)$ denotes the vectorization operation, $C_S^{l_S}$ denotes the correlation between different channels in the student hidden layer l_S and $C_T^{l_T}$ denotes the correlation between different channels in the teacher hidden layer l_T.

As verified in previous work on neural style transfer [7], the correlation between channels C^l is blind to the global arrangement of images. To account for spatial structure in images, we also consider the correlation between spatial locations. We use the Gram matrix $S^l \in \mathbb{R}^{M_l \times M_l}$ to denote the correlation between spatial locations, where S^l_{ij} is defined as the inner product between the feature activations on location i and location j:

$$S^l_{ij} = \sum_{k=1}^{N_l} F^l_{ki} F^l_{kj}. \tag{3}$$

Similar to the correlation between channels, we transfer the correlation between spatial locations from the teacher to the student with the following L_2 loss function:

$$L_{\text{Spatial}} = \| vec(norm(S_S^{l_S})) - vec(norm(S_T^{l_T})) \|_2, \tag{4}$$

where $norm(\cdot)$ and $vec(\cdot)$ have the same definition as above, $S_S^{l_S}$ denotes the correlation between different spatial locations in the student hidden layer l_S and $S_T^{l_T}$ denotes the correlation between different spatial locations in the teacher hidden layer l_T.

When transferring $C_T^{l_T}$ to $C_S^{l_S}$, we assume that the student hidden layer has the same number of channels or feature maps with the teacher hidden layer. When transferring $S_T^{l_T}$ to $S_S^{l_S}$, we assume that the feature map of the student hidden layer has the same spatial dimensions with the teacher hidden layer. The two requirements can be easily satisfied in practice as we will see in our experiments. If any requirement is not satisfied, we can add a regression layer on top of $C_S^{l_S}$ or $S_S^{l_S}$ such that the output matches the size of $C_T^{l_T}$ or $C_S^{l_S}$. A similar strategy has also been employed in FitNets [8]. We do not focus on this in this paper and leave this as future work.

We denote L_{Know} as the loss function used for transferring the knowledge from the teacher network to the student network. Based on our above explanation, L_{Know} can be L_{Channel} alone, L_{Spatial} alone or the average of L_{Channel} and L_{Spatial}. L_{Know} provides supervision

for an intermediate layer in the student network[1]. In addition, we supervise the class probability distribution output by the student network using the cross entropy with ground truth labels. We can also combine the standard cross entropy loss function with the loss function proposed in Knowledge Distillation (KD) [9]. To be more concrete, we use the following loss function L_{Class} to guide the class probability distribution output by the student network:

$$L_{\text{Class}} = \alpha L_{\text{GT}} + (1 - \alpha) L_{\text{KD}} \tag{5}$$

where L_{GT} denotes the cross entropy with ground truth labels, L_{KD} is proposed by KD [3] and denotes the cross entropy with the soft target distribution produced by the teacher, and α [0,1] controls the weight between L_{GT} and L_{KD}. Note that the cross entropy in L_{KD} is scaled by the square of the temperature used for computing the soft target distribution, as suggested by KD [3].

Now we explain our training scheme of the student network. As illustrated in the top figure in Fig. 2, L_{Know} is employed to supervise an intermediate student layer. It influences the first layer up to the guided layer in the student network. L_{Class} guides the class probability distribution output by the student network and influences the whole student network.

We can train the student network in a single-stage fashion or a two-stage fashion. For the single-stage training, we use the weighted sum of L_{Know} and L_{Class} as the loss function, which is defined as follows,

$$L_{\text{Single}} = L_{\text{Know}} + \beta L_{\text{Class}}, \tag{6}$$

where β is the weight parameter. The two-stage training fashion is similar to FitNets [4]. We first train the student network from the first layer up to the guided layer with L_{Know}. At the second stage, we train the whole network with L_{Class}.

3.2 Self-Guided Learning

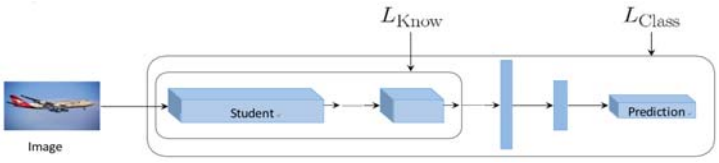
In previous work [4, 5], when they transfer activations [4] or attention maps [5] from an intermediate teacher layer to an intermediate student layer, all the neurons in the guided student layer are forced to be supervised by the knowledge from the teacher. To give the student more flexibility and enable student-centric learning, we propose to give the student opportunity to learn by itself in the form of additional self-taught neurons.

As shown in Fig. 1, self-taught neurons refer to neurons that do not receive knowledge from the teacher and are learned by the student itself from scratch using only the supervisory signals of the data. For self-guided learning, we append the guided student layer with additional self-taught neurons and concatenate the intermediate representations learned by the newly added neurons and original neurons in the layer, before passing them to the next layer. We can add self-taught neurons to more than one guided student layer if there are multiple intermediate student layers receiving knowledge from the teacher.

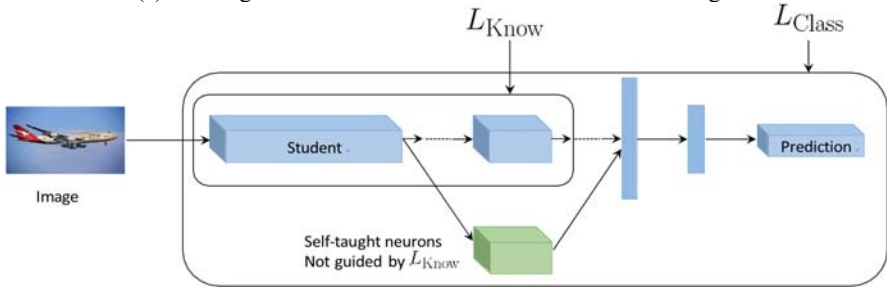
We illustrate the training scheme of the student network with self-taught neurons in the bottom figure in Fig. 2. The training scheme is similar to that of the student network without self-taught neurons. The L_{Know} is employed to supervise an intermediate student layer and the L_{Class} guides the class probability distribution output by the student network and influences the whole student network. The only difference is that L_{Know} does not supervise the self-taught neurons in the guided student layer and only supervise other neurons

[1] We can also supervise multiple student layers using the knowledge from multiple teacher layers. This will require us to select multiple pairs of student/teacher layers manually and we leave this for future study.

in that layer. L_{Know} influences the first layer up to the guided layer in the student network, except self-taught neurons. Similarly, we can train the student network, following the single-stage fashion or the two-stage fashion detailed above.



(a) Training scheme of the student network without self-taught neurons.



(b) Training scheme of the student network with self-taught neurons.

Fig. 2. Training scheme of the student network.

Note that when combined with KD, *i.e.*, when the parameter α in Eq. (5) is smaller than 1, all the student networks including the self-taught neurons are guided by the knowledge of the teacher in the form of soft target distribution. But the self-taught neurons never receive knowledge directly from intermediate teacher layers. Guiding the whole student network with soft target distribution still allows the self-taught neurons to learn its own intermediate representations.

It is also worthwhile to mention that the idea of self-taught neurons is generic and applicable no matter the form of the knowledge. We can combine self-guided learning with correlation-based learning or any other forms of knowledge.

4. EXPERIMENTAL EVALUATION

In this section, we provide quantitative evaluation of our proposed approaches correlation-based learning and self-guide learning for the task of model compression. We use the TensorFlow as a tool to implement the networks and train them on a Nvidia GTX-1080.

4.1 Datasets and Settings

We conduct experiments on two benchmark datasets: CIFAR-10 [19] and CIFAR-100 [11]. The CIFAR-10 dataset contains 60,000 color images of size 32×32, which can be divided into 10 categories and 6,000 images for each category. The CIFAR-100 dataset contains 60,000 color images of size 32×32 in 100 categories, with 600 images per category.

We experiment with network architectures based on the widely-used VGG network [21]. The detailed network architectures used in our experiments are shown in Table 1. The number of their parameters is shown in Table 2. The teacher-augmented network and student-augmented network are used in our experiments regarding self-guided learning. Please refer to Section 4.3 for more details. Some layers in Table 1 are shown in boldface. They are the selected intermediate layers to provide knowledge (teacher, teacher-augmented) or to receive knowledge (student, student-augmented). These layers have more parameters than other layers and have a greater impact on the final prediction, which can effectively guide the training of student network.

Table 1. Network architectures (shown in columns).

Network Architecture			
Teacher	Student	Teacher-Augmented	Student-Augmented
11 weight layers	5 weight layers	11 weight layers	5 weight layers
conv3-64 conv3-64	conv3-64	conv3-64 conv3-64	conv3-64
maxpool			
conv3-128 conv3-128	conv3-128	conv3-128 conv3-128	conv3-128
maxpool			
conv3-256 conv3-256 conv3-256	conv3-256	conv3-256 conv3-256 conv3-256	conv3-256
maxpool			
conv3-512 conv3-512 conv3-512	**conv3-512**	**conv3-1024** conv3-512 conv3-512	**conv3-1024**
global average pooling			
fc-n (n = no. of classes)			
softmax			

The convolutional layer parameters are denoted as "conv<receptive field size>-<number of channels>". The ReLU function and batch normalization layers are not shown for brevity.

Table 2. Number of parameters.

Dataset	Teacher	Student	Compression ratio	Teacher-Augmented	Student-Augmented	Compression ratio
CIFAR-10	7.65M	1.56M	20.39%	11.19M	2.75M	24.58%
CIFAR-100	7.69M	1.61M	20.93%	11.23M	2.84M	25.29%

4.2 Evaluation of Correlation-based Learning

In this section, we evaluate the performance of correlation-based learning and show the results in Table 3. We consider three baselines: transferring activations [4], transferring attention maps [5] and KD [11]. For correlation-based learning, we consider three cases: using the correlation between channels alone (Channel, $L_{\text{Know}} = L_{\text{Channel}}$), using the correlation between spatial locations alone (Spatial, $L_{\text{Know}} = L_{\text{Spatial}}$) and using both of them (Both, $L_{\text{Know}} = \frac{1}{2}(L_{\text{Chanel}} + L_{\text{Spatial}})$). For each approach, we have tried training the student work with

the single-stage training fashion and the two-stage stage training fashion. When following the single-stage training fashion, we set β to 0.1. We have tried setting the value of α to 1 or 0.3. When α equals to 1, the student network does not take advantage of the soft target distribution output by the teacher and the performance is only based on our correlation-based learning approach. α equaling to 0.3 means that our approach correlation-based learning is combined with KD to obtain the final performance. In Table 3, 'Single' and 'Two' means training the student network following the single-stage and two-stage training fashion respectively.

Table 3. Accuracy on CIFAR-10 and CIFAR-100 for correlation-based learning and previous approaches.

Approach	CIFAR-10				CIFAR-100			
	Single	Single+KD	Two	Two+KD	Single	Single+KD	Two	Two+KD
Channel	90.68	90.27	91.03	91.62	67.10	68.37	66.99	69.25
Spatial	**91.27**	**90.92**	90.77	91.27	66.88	68.36	67.41	68.73
Both	89.78	89.26	**91.11**	**91.67**	**67.11**	**68.59**	**67.70**	69.21
Activation	90.37	90.06	91.04	91.32	66.58	67.23	67.48	**69.38**
Attention	90.52	90.58	90.40	90.97	66.99	67.77	66.46	68.78
KD	89.73				67.36			
Teacher	92.02				68.68			
Student	89.18				64.66			

The highest performance under each setting is shown in boldface. See text for more details.

The suffix '+KD' means training the student network with $\alpha = 0.3$. If there is no suffix '+KD', the student network is trained with $\alpha = 1$.

From Table 3 we can see that our correlation-based learning approach consistently outperforms transferring activations [4] and attention maps [5] under almost all the settings. Particularly, we find that combining the two types of correlations yields better results than using only one type of correlation in most cases. Moreover, two-stage training consistently performs better than single-stage training. Also, combining with KD always gives us additional improvement.

Notably, on CIFAR-10, correlation-based learning combined with KD can achieve the accuracy of 91.67%, which is very close to 92.02%, the accuracy of the teacher network. On CIFAR-100, we can see that our student network can obtain the accuracy 69.25%, which is higher than 68.68%, the accuracy of the teacher network.

4.3 Evaluation of Self-Guided Learning

Self-guided learning can be combined with any form of knowledge. In our experiments, we consider two forms of knowledge: the correlation between channels proposed by us and the direct activation matching proposed by FitNets [4]. The results are summarized in Table 4.

The experiments involve all the four networks shown in Table 1. In Table 4, we show three set- tings: 'Original', 'Augmented' and 'Augmented+Self-Guided'. 'Original' refers to transferring the knowledge from the teacher network to the student network. 'Augmented' refers to transferring the knowledge from the teacher-augmented network to the

Table 4. Accuracy on CIFAR-100 for self-guided learning.

Knowledge	Setting	CIFAR-100	
		Single	Single+KD
Channel	Original	67.10	68.37
	Augmented	67.67	69.10
	Augmented+Self-Guided	**68.21**	**69.58**
Activation	Original	66.58	67.23
	Augmented	**68.04**	68.03
	Augmented+Self-Guided	67.89	**68.40**
Student-Augmented	66.26		

The highest performance among 'Original', 'Augmented' and 'Augmented+Self-Guided' is shown in boldface. See text for more details.

student-augmented network. 'Original' and 'Augmented' do not involve our self-guided learning approach. Under these two settings, we are simply transferring the correlation between channels or the activations from the teacher to the student. 'Augmented+Self-Guided' means that we transfer the knowledge from the teacher network to the student-augmented network. Under 'Augmented+Self-Guided', the intermediate teacher layer[2] only has 512 channels but the intermediate student layer[3] has 1024 channels, of which 512 channels are supervised by the knowledge from the teacher and the other 512 channels are self-taught neurons.

Comparing 'Augmented+Self-Guided' and 'Original', we can see that by adding additional self- taught neurons to the intermediate student layer, we can improve the accuracy by more than 1% on CIFAR-100, whether the knowledge provided by the teacher is the correlation between channels or the activation. Note that 'Augmented+Self-Guided' and 'Original' use the same teacher network and the only difference is whether the student have additional self-taught neurons.

One may argue that the improvement of the performance comes from the increase of the model capacity due to the added self-taught neurons, so we also compare 'Augmented+Self-Guided' to 'Augmented'. They both use the student-augmented network and the difference is that 'Augmented' do not have self-taught neurons. Under 'Augmented', all the 1024 channels in the intermediate student layer are supervised by the knowledge from the teacher-augmented network. But under 'Augmented+Self-Guided', only 512 channels in the intermediate student layer are supervised by the knowledge from the teacher network. As shown in Table 4, 'Augmented+Self-Guided' outperforms 'Augmented' in most cases. Notably, when using the correlation between channels as the form of knowledge, which is proposed by us, we can achieve the highest performance 69.58%, which is higher than 68.68%, the accuracy of the teacher network, by nearly 1%. Note that the student (student-augmented network) is only 36% of the size of the teacher (teacher network) in this case. The student network inherits the knowledge of the teacher network and adds self-taught neurons, which make the student network not only retains the performance of the teacher network, but also has more accurate predictions for specialized fields.

5. CONCLUSION

Towards student-centric learning, we have proposed two approaches to allow the student to have more flexibility during training: correlation-based learning and self-guided

[2] The layer shown in boldface under the column 'Teacher' in Table 1.
[3] The layer shown in boldface under the column 'Student-Augmented' in Table 1.

learning. We validate our proposed approaches with extensive experimental results and show that our approaches can train a smaller and shallower student network with only 5 layers that outperforms a larger and deeper teacher network with 11 layers by nearly 1% on CIFAR-100. We believe the idea of student-centric learning and our proposed approaches will help further advance knowledge transfer from a teacher to a student.

ACKNOWLEDGMENTS

This work was funded by Science and Technology Plan of China Administration of Market Supervision [2019MK003], and supported by the Fundamental Research Funds for the Central Universities (2019JBM025).

REFERENCES

1. J. Ba and R. Caruana, "Do deep nets really need to be deep?" in *Advances in Neural Information Processing Systems*, 2014, pp. 2654-2662.
2. C. Bucilua, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2006, pp. 535-541.
3. G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," arXiv preprint arXiv:1503.02531, 2015.
4. A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: Hints for thin deep nets," arXiv preprint arXiv:1412.6550, 2014.
5. S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," arXiv preprint arXiv:1612.03928, 2016.
6. J. Liu, C. Sun, X. Xu, B. Xu, and S. Yu, "A spatial and temporal features mixture model with body parts for video-based person re-identification," *Applied Intelligence*, Vol. 49, 2019, pp. 3436-3446.
7. L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2414-2423.
8. Y. Wang, D. Ramanan, and M. Hebert, "Growing a brain: Fine-tuning by increasing model capacity," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2471-2480.
9. S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," arXiv preprint arXiv: 1510.00149, 2015.
10. B. Hassibi, D. G. Stork, *et al.*, "Second order derivatives for network pruning: Optimal brain surgeon," *Advances in Neural Information Processing Systems*, 1993, pp. 164-164.
11. A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Technical Report, University of Toronto, 2009.
12. S. J. Hanson and L. Y. Pratt, "Comparing biases for minimal network construction with back-propagation," in *Advances in Neural Information Proceedings*, Vol. 1, 1989,

pp. 177-185.
13. X. Xu, X. Wang, and K. M. Kitani, "Error correction maximization for deep image hashing," in *Proceedings of British Machine Vision Conference*, 2018, pp. 1-12.
14. S. Arora, A. Bhaskara, R. Ge, and T. Ma, "Provable bounds for learning some deep representations," in *Proceedings of the 31st International Conference on Machine Learning*, Vol. 32, 2014, pp. 584-592.
15. Y. Gong, L. Liu, M. Yang, and L. Bourdev, "Compressing deep convolutional networks using vector quantization," arXiv preprint arXiv:1412.6115, 2014.
16. K. Hwang and W. Sung, "Fixed-point feedforward deep neural network design using weights +1, 0, and −1," in *Proceedings of IEEE Workshop on Signal Processing Systems*, 2014, pp. 1-6.
17. Z. Lin, M. Courbariaux, R. Memisevic, and Y. Bengio, "Neural networks with few multiplications," arXiv preprint arXiv:1510.03009, 2015.
18. M. Courbariaux, Y. Bengio, and J.-P. David, "Training deep neural networks with low precision multiplications," arXiv preprint arXiv:1412.7024, 2014.
19. M. Kim and P. Smaragdis, "Bitwise neural networks," arXiv preprint arXiv:1601.06071, 2016.
20. M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "Xnor-net: Imagenet classification using binary convolutional neural networks," in *Proceedings of European Conference on Computer Vision*, 2016, pp. 525-542.
21. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.

Hong-Ji Wang is a Master student at the School of Computer and Information Technology, Beijing Jiaotong University. He obtained B.S. in Computer Science from Beijing Jiaotong University. His research interests lie in recommendation system and CTR prediction

Xiang Xu is a Ph.D. candidate at the School of Computing Science, Simon Fraser University. He obtained B.S. in Electrical and Computer Engineering from Carnegie Mellon University. His research interests lie in computer vision and machine learning, with a particular focus on 3D understanding that treats human as 3D sensor for modeling human-object relations.

Bao-Min Xu is a Professor in School of Computer and Information Technology, Beijing Jiaotong University. He received the Ph.D. degree in Computer Science in 2000 from the Institute of Computing Technology at the Chinese Academy of Sciences, China. His research interests include distributed computing and computer vision. Dr. Xu has published nearly 50 papers, of which were retrieved by SCI and EI Journal, in the field of distributed computing and cloud computing.

Shuang-Yuan Yu is a Professor in School of Computer and Information Technology, Beijing Jiaotong University, China. Her received the Master degree in Computer Science in 1990 from the I Beijing Jiaotong University. Her research interests include Distributed Computing and Computer Vision.

Quan-Xin Wang is a Lecture in Beijing Jiaotong University Haibin College, China. Her Graduated from the Beijing Normal University in 2010. Her research interests include programming language and database.