The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Adversarial Learning of Portable Student Networks

Yunhe Wang,[1,3] **Chang Xu,**[2] **Chao Xu,**[1,3] **Dacheng Tao**[2]
[1]Key Laboratory of Machine Perception (MOE), School of EECS, Peking University, China
[2]UBTECH Sydney AI Centre, SIT, FEIT, University of Sydney, Australia
[3]Cooperative Medianet Innovation Center, Peking University, China
wangyunhe@pku.edu.cn, c.xu@sydney.edu.au,
xuchao@cis.pku.edu.cn, dacheng.tao@sydney.edu.au

Abstract

Effective methods for learning deep neural networks with fewer parameters are urgently required, since storage and computations of heavy neural networks have largely prevented their widespread use on mobile devices. Compared with algorithms which directly remove weights or filters for obtaining considerable compression and speed-up ratios, training thin deep networks exploiting the student-teacher learning paradigm is more flexible. However, it is very hard to determine which formulation is optimal to measure the information inherited from teacher networks. To overcome this challenge, we utilize the generative adversarial network (GAN) to learn the student network. In practice, the generator is exactly the student network with extremely less parameters and the discriminator is used as a teaching assistant for distinguishing features extracted from student and teacher networks. By simultaneously optimizing the generator and the discriminator, the resulting student network can produce features of input data with the similar distribution as that of features of the teacher network. Extensive experimental results on benchmark datasets demonstrate that the proposed method is capable of learning well-performed portable networks, which is superior to the state-of-the-art methods.

Introduction

As one of recent most effective tools for implementing machine intelligence tasks, deep neural networks, especially convolutional neural networks (CNNs), have successfully addressed a number of real world problems, *e.g.*, image classification (Simonyan and Zisserman 2015; Krizhevsky, Sutskever, and Hinton 2012), visual detection and segmentation (Ren et al. 2015; Long, Shelhamer, and Darrell 2015), audio recognition and analysis (Liang, Jiang, and Hauptmann 2017), *etc*. Owing to the large amount of accessible training data and computational power of GPUs, a series of impressive CNNs have been developed to continuously boost the performance of deep learning. For instance, the ResNet-50 (He et al. 2015) obtained an about 3.8% top-5 error rate on the ILSVRC 2012 dataset (Russakovsky et al. 2015), which is slightly lower than that of human eyes.

Besides work stations and PC with GPU cards, mobile devices (*e.g.*, telephone, micro robot) also look forward to the applications of CNNs. However, launching sophisticated CNNs on these low-configure devices is almost impossible since massive storage and a number of floating number multiplications would be consumed. For instance, over $232MB$ of memory and over 7.24×10^8 multiplications are demanded for processing one image using AlexNet (Krizhevsky, Sutskever, and Hinton 2012), which cannot be tolerated by these devices. Therefore, portable deep models with similar accuracies are urgently expected.

To this end, there are a variety of methods have been proposed for compressing convolutional neural networks such as vector quantization (Gong et al. 2014), weight matrix decomposition (Denton et al. 2014), encoding (Chen et al. 2015), and pruning (Wang et al. 2016; Han, Mao, and Dally 2016). Wherein, weight pruning has shown an extraordinary performance on most of the benchmark deep models. In specific, (Han, Mao, and Dally 2016) showed that over 80% subtle weights can be removed without affecting performance of the original networks. (Wang et al. 2016) further pointed out that the redundancy can exist in both large and small weights and explored an effective compression method in the frequency domain.

Although, these sparsity based methods can achieve considerable compression and speed-up ratios by preserving the accuracies, specialized hardwares are often required for efficient inference. Hence, another more straightforward strategy is very popular recently, *i.e.*, directly learning a portable network (*student network*) with fewer parameters to inherit valuable properties of the original network (*teacher network*) (Hinton, Vinyals, and Dean 2015; Romero et al. 2014), which has the similar purpose to transfer learning (Luo et al. 2014; 2017). Considering the consensus that the performance of the network usually improves with the increasing of network depth (Ba and Caruana 2014), the student network of the thinner and deeper architecture could also achieve similar accuracy. Various techniques have been exploited to measure and decrease the discrepancy between student network and teacher network, such as minimizing the Euclidean distance between features extracted from hidden layers of the two networks (Ba and Caruana 2014), inheriting the classification results (Hinton, Vinyals, and Dean 2015), and transferring feature information from intermediate layers (Romero et al. 2014). These approaches have investigated the consistency between stu-

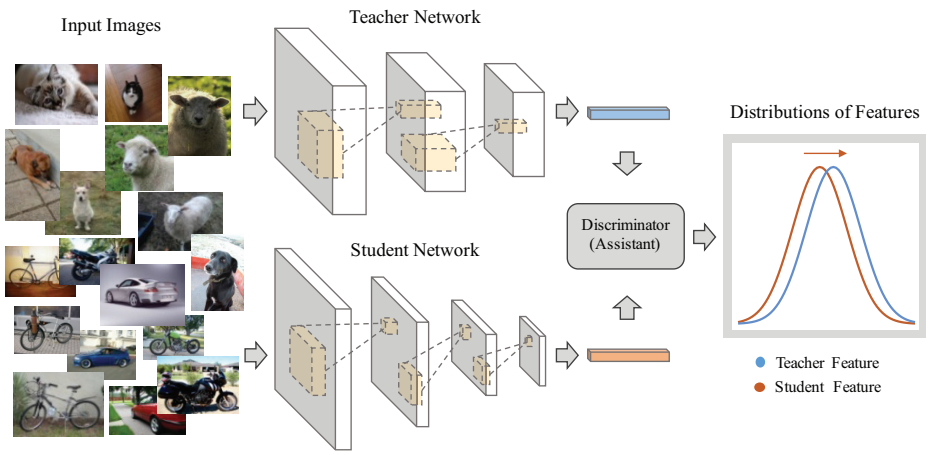


Figure 1: The diagram of the proposed method for learning portable deep neural networks by exploiting GAN. The top line is the teacher network with heavy parameters and the bottom line is the student network (Generator) with significantly fewer parameters. The teaching assistant (Discriminator) is employed on features extracted from two networks with the same input data, which makes their distributions similar in the same feature space.

dent and teacher networks from different aspects. It is difficult to determine which measurement is optimal, and we also need to ask whether all possible measurements have been noted. Instead of insisting on some particular measurement, we expect a more comprehensive and thorough evaluation on the consistency between student and teacher networks.

In this paper, we suggest to develop a teaching assistant network to identify the difference between features generated by student and teacher networks. The capability of the teaching assistant network will be improved through continuous optimization. If no matter how teaching assistant network transforms the features from student and teacher networks, they still cannot be distinguished from each other, and then the student network is treated as a satisfying successor of the teacher network. The teaching assistant network and student networks have naturally formed the generative adversarial networks (GANs), where the student network plays as a generator and the teaching assistant network aims to distinguish the features generated by student network and the pre-trained teacher network respectively. By simultaneously minimizing the classification error of input images themselves and the loss in GANs, the optimal student network can be discovered with the help of the teacher network. Experiments conducted on benchmark datasets and models demonstrate the superiority of the proposed algorithm over the state-of-the-art methods for learning portable deep neural networks.

Related Works

We first briefly introduce related works on learning convolutional neural networks of fewer parameters. Based on their techniques and motivations, these methods can be divided into two categories.

Network Trimming

Network trimming is the most common scheme for compressing deep neural networks, which aims to remove redundancy in the original network and generate a network with less memory usage and computational complexity. (Gong et al. 2014) exploited vector quantization to use a cluster center to represent a set of similar weights. (Denton et al. 2014) regarded weights of fully connected layers as low-rank matrices and decomposed them using the singular value decomposition approach. Besides excavating similar weights or filters, 32-bit floating numbers are over-refined. Therefore, (Rastegari et al. 2016; Courbariaux and Bengio 2016; Arora et al. 2014) explored binary networks, whose weights are -1/1, or -1/0/1. In addition, (Han, Mao, and Dally 2016) employed conventional data compression techniques such as pruning (Han et al. 2015), quantization, and Huffman coding to obtain a much higher compression ratio. Subsequently, (Wang et al. 2016) further converted convolution filters into the frequency domain to excavate more redundancy and explored a novel convolution operation, thereby producing state-of-the-art CNNs compression.

Although tremendous efforts have been taken to develop the aforementioned algorithms, compressed networks produced by them are different to their original ones, and need specialized hardwares (*e.g.*, fixed point multiplier), which will increase design and development costs of mobile devices.

Student Networks Learning

Besides applying compression algorithms to directly process the heavy networks, there are some works investigating the intrinsic information captured by original networks in order to learn thinner and deeper networks. (Ba and Caruana 2014) attempted to minimize the difference of features extracted from a deeper network and the heavy teacher network. (Hinton, Vinyals, and Dean 2015) constructed a thinner neural

network and then made its outputs of the softmax layer similar to those of the teacher network to maintain the performance. (Romero et al. 2014) minimized the difference between features of an arbitrary layer in the student network and a given layer in its teacher network, which enables the thinner and deeper student network to own an acceptable accuracy drop. (McClure and Kriegeskorte 2016) proposed to minimize the pairwise distance of samples between the student network and the teacher network for having a more robust performance. In addition, a number of techniques have been developed to relax the restrict assumptions, *e.g.*, attention transfer (Zagoruyko and Komodakis 2016) and knowledge pre-regression (Wang, Deng, and Wang 2016). However, they often independently treat each input image and neglect the whole distribution of examples. (Wang, Deng, and Wang 2016) employs maximum mean discrepancy (MMD) to minimize feature distributions of teacher and student networks by exploiting linear and nonlinear kernel functions. But the optimal kernel functions in MMD are difficult to determine in practice. Moreover, (You et al. 2017) simultaneously utilized multiple teacher networks for learning a more accurate student network. (Wang et al. 2017b) proposed to discard redundancy in feature maps produced by numerous filters, and then reconstruct a compact network with an acceptable accuracy reduction.

Compared with network trimming approaches, portable networks generated by the teacher-student paradigm are much more flexible since they do not need any additional supports for implementing online inference. However, performance of these student networks is usually a bit lower than those of teacher networks since they cannot comprehensively and accurately measure the consistency between student and teacher networks. Therefore, a more effective scheme for inheriting useful information from teachers is imperative.

Learning Student Networks with GAN

This section analyzes which information of teacher networks should be inherited and proposes a novel teacher-student learning framework by exploiting generative adversarial networks.

Teacher-Student Interactions

In general, a CNN is learned on a relatively large dataset with a large number of examples accompanied with ground-truth labels. Here we follow the settings in (Hinton, Vinyals, and Dean 2015; Romero et al. 2014) to examine the compression task under the image classification problem, which is one of the most widely applications of CNNs.

Denote the original pre-trained convolutional neural network (teacher network) as $\mathcal{N}_T$ and the desired portable network (student network) as $\mathcal{N}_S$. Commonly, $\mathcal{N}_S$ has more convolutional layers but fewer parameters compared with $\mathcal{N}_T$. Let $\mathcal{X}$ denote the example sapce and $\mathcal{Y}$ is its corresponding k-label space. Given a labeled training set with n samples, $\{(\mathbf{x}^1, \mathbf{y}^1), (\mathbf{x}^2, \mathbf{y}^2), ..., (\mathbf{x}^n, \mathbf{y}^n)\}$, we denote features (*i.e.*the input data of the softmax layer) of $\mathbf{x}^i$ extracted by the two networks as $z_T^i = \mathcal{N}_T(\mathbf{x}^i)$ and $z_S^i = \mathcal{N}_S(\mathbf{x}^i)$, respectively. The loss function of the conventional softmax is

$$J(\theta) = -\frac{1}{n}\left[\sum_{i=1}^{n}\sum_{j=1}^{k}\mathbf{1}\{\mathbf{y}^i = j\}\log\frac{e^{\theta_j^\top z_T^i}}{\sum_{l=1}^{k} e^{\theta_l^\top z_T^i}}\right], \quad (1)$$

where θ_l is the parameter vector for the l-th category, $\mathbf{1}\{\cdot\}$ is the indicator function. By adopting the feed forward and back propagation strategies, the original teacher network can capture redundant information from the training set and yields a satisfying model with considerable performance.

As for the student network of significantly fewer neurons and weights, much more efforts are required to improve its performance to that of the teacher network. Therefore, we should excavate some useful information from the teacher network as guidances for helping us to train the student network. A straightforward idea is to encourage features from student and teacher networks to be similar, so that input data can be accurately recognized by the student network through minimizing Fcn. 1. (Ba and Caruana 2014; Chen et al. 2015) proposed using the following objective function to learn the portable student network:

$$\mathcal{L}(\mathcal{N}_S) = \frac{1}{n}\sum_{i=1}^{n}\left[\mathcal{H}(o_S^i, \mathbf{y}^i) + \frac{\lambda}{2}||z_S^i - z_T^i||_2^2\right], \quad (2)$$

where o_S^i is the output of the classier given the input feature z_S^i, *i.e.*, $o_S^i = e^{\theta^\top z_S^i}/||e^{\theta^\top z_S^i}||_1$, $\mathcal{H}(\cdot,\cdot)$ is the cross entropy loss for guaranteeing the performance of the student network, and λ is a weight parameter for balancing the classification accuracy and the difference between features extracted from teacher and student networks.

However, there are significant differences between teacher and student networks, *e.g.*, depth, number of weights, features extracted by $\mathcal{N}_S$ cannot be easily similar to those extracted by $\mathcal{N}_T$. In order to provide more powerful priori, (Hinton, Vinyals, and Dean 2015) proposed distilling knowledge from the classification result with a softening parameter τ:

$$\mathcal{L}(\mathcal{N}_S) = \frac{1}{n}\sum_{i=1}^{n}\left[\mathcal{H}(o_S^i, \mathbf{y}^i) + \lambda\mathcal{H}(\tau(o_S^i), \tau(o_T^i))\right], \quad (3)$$

where the second term calculates the cross entropy loss between outputs of student and teacher, and

$$o_S^i = \frac{e^{\theta_S^\top z_S^i}/\tau}{||e^{\theta_S^\top z_S^i}/\tau||_1}, \quad o_T^i = \frac{e^{\theta_T^\top z_T^i}/\tau}{||e^{\theta_T^\top z_T^i}/\tau||_1}, \quad (4)$$

θ_S and θ_T are classifier parameters of student and teacher networks, respectively. $\tau > 1$ is a temperature parameter for the softening manipulation. Since the above two functions extract information between labels, this softening strategy has shown extraordinary effect for helping the student network to inherit rich information from the teacher network. In addition, (Romero et al. 2014) utilized another fully connected layer to connect feature maps of student and teacher networks from different layers for making them similar. (McClure and Kriegeskorte 2016) proposed to keep the pairwise distance values of samples between student and teacher networks at some intermediate layer.

In fact, all these methods agree that the teacher network can provide valuable information for helping us to learn portable student network. Regularizations or measurements used in existing methods are mainly fixed from different perspectives. In this paper, we attempt to explore a more comprehensive way to learn portable networks by minimizing the discrepancy between distributions of features extracted from student and teacher networks. Thus, we propose to use the generative adversarial network (GAN) for implementing this meaningful task.

GAN for Student Network Learning

The recently proposed generative adversarial network (GAN, (Goodfellow et al. 2014; Mirza and Osindero 2014; Wang et al. 2017a)) consists of a discriminator D and a generator G. The training process of GANs can be regarded as a two players minimax game, which has been successfully applied into a number of computer vision applications such as image super-resolution (Ledig et al. 2016), visual generation (Reed et al. 2016), style transfer (Isola et al. 2016), *etc*.

In the general GAN, the generator G maps an input noise vector z with a specific distribution to the desired data y, *i.e.*, $G : z \rightarrow y$, and the task of the discriminator D is to distinguish the original data from the synthetic data $G(z)$, the objective function can be formulated as

$$\begin{aligned}\mathcal{L}_{GAN}(z, y) =& \mathbb{E}_{y\sim p_{data}(y)}[\log D(y)] \\ &+ \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z))],\end{aligned} \tag{5}$$

where the generator will be adjusted according to the training error produced by D using the back propagation strategy, and the optimal generator is

$$G^* = \arg\min_G \max_D \mathcal{L}_{GAN}. \tag{6}$$

In fact, the generator G is a general deep neural network that can be investigated in a number of real-world applications. Therefore, it is reasonable for us to use a generator to implement the visual classification task. In specific, the generator G will be regarded as a student network, which maps the given image $\mathbf{x}^i$ to its ground-truth label $\mathbf{y}^i$.

As mentioned above, the information of the teacher network can be distilled from two aspects, *i.e.*, features of input sample z_T^i (Ba and Caruana 2014; Romero et al. 2014), and output values of classifiers o_T^i or $\tau(o_T^i)$ (Hinton, Vinyals, and Dean 2015). In order to inherit information from the teacher network $\mathcal{N}_T$ as much as possible, we divide the generator G (*i.e.*, the student network $\mathcal{N}_S$) into two parts, the first part extracts the feature of input data and the second part outputs the classification result, *i.e.*,

$$z_S^i = G_1(\mathbf{x}^i), \quad o_S^i = G_2(z_S^i), \tag{7}$$

Then, we use the following objective function to pursue the consistency between student network and teacher network $\mathcal{N}_T$:

$$\begin{aligned}\mathcal{L}_{GAN} = \frac{1}{n}\sum_{i=1}^{n}\mathcal{H}(o_S^i, \mathbf{y}^i) + \gamma\frac{1}{n}\sum_{i=1}^{n}\Big[\Big(\log(D(z_T^i)) \\ + \log(1 - D(z_S^i))\Big)\Big],\end{aligned} \tag{8}$$

Algorithm 1 Learning portable DNNs by exploiting GAN.

Input: A given neural network $\mathcal{N}_T$ and its train dataset $\mathcal{X}$ with n instances and $\mathcal{Y}$ is the corresponding k-label set, parameters: λ, γ, and τ.

1: Manually initialize a GAN with a Generator $G = [G_1, G_2]$ and a Discriminator D, where the number of parameters in G is significantly fewer than that in $\mathcal{N}_T$;
2: **repeat**
3: Randomly select an instance $\mathbf{x}$ and its label $\mathbf{y}$;
4: Employ the teacher network: $[z_T, o_T] \leftarrow \mathcal{N}_T(\mathbf{x})$;
5: Employ the generator: $z_S \leftarrow G_1(\mathbf{x})$, $o_S \leftarrow G_2(z_S)$;
6: Calculate $D(\mathbf{x}_T)$ and $D(\mathbf{x}_S)$, and update weights in the discriminator$D$ accordingly;
7: $\tau(o_S) \leftarrow \frac{e^{\theta_S^\top z_S/\tau}}{||e^{\theta_S^\top z_S/\tau}||_1}$, $\tau(o_T) \leftarrow \frac{e^{\theta_T^\top z_T/\tau}}{||e^{\theta_T^\top z_T/\tau}||_1}$;
8: Calculate the loss function $\mathcal{L}_{GAN}$ (Fcn. 9);
9: Update weights in G_1 and G_2 using gradient descent;
10: **until** convergence

Output: The portable deep neural network $\mathcal{N}_S = G^*$.

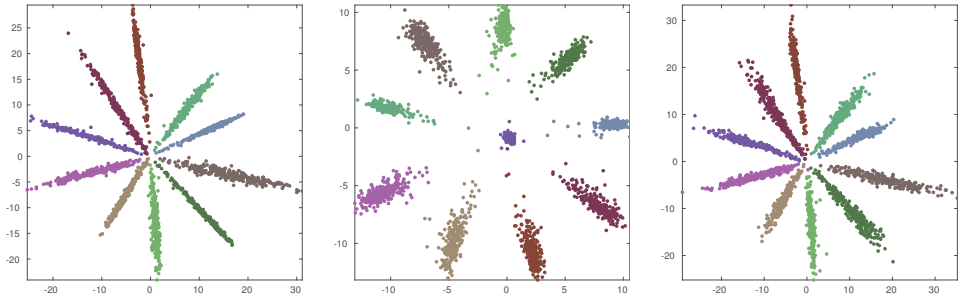
where γ is the weight parameter for seeking the trade-off of two different terms.

Compared with Fcn. 2, the first term in Fcn. 8 minimizes the cross entropy loss of classifier outputs to maintain the performance of the student network, while the second term is to expect the features extracted by student and teacher networks are indistinguishable from each other (*i.e.*, they cannot be separated by a sophisticated classifier). Therefore, by simultaneously optimizing these two objectives, image features generated by the student network $\mathcal{N}_S = [G_1, G_2]$ will follow the similar distribution as that of features generated by the original teacher network as shown in Fig. 1, which is obviously beneficial to the training process of the student network.

Note that the knowledge of teacher network has been investigated from two different aspects, *i.e.*, feature (Ba and Caruana 2014; Romero et al. 2014) and classification output (Hinton, Vinyals, and Dean 2015). In fact, these two strategies can be integrated into a more compact formulation, and Fcn. 8 is rewritten as:

$$\begin{aligned}\mathcal{L}_{GAN} =& \frac{1}{n}\sum_{i=1}^{n}\Big[\mathcal{H}(o_S^i, \mathbf{y}^i) + \lambda\mathcal{H}(\tau(o_S^i), \tau(o_T^i))\Big] \\ &+\gamma\frac{1}{n}\sum_{i=1}^{n}\Big[\log(D(z_T^i)) + \log(1 - D(z_S^i))\Big],\end{aligned} \tag{9}$$

where the first term in Fcn. 9 is exactly Fcn. 3 which captures useful information from outputs of the teacher network , and the second term illustrates the significance of "teaching assistant" to enable teacher and student networks to generate features following the same distributions. Classical mini-batch strategy will be applied to optimize these networks. Alg. 1 summarizes the detailed procedure of the proposed approach for learning portable student models. After training the generative network through the given dataset $\mathcal{X}$ and $\mathcal{Y}$, the resulting optimal generator G^* is the compressed portable deep neural network.



(a) accuracy = 99.2% (b) accuracy = 97.2% (c) accuracy = 99.1%

Figure 2: Visualization results of different networks trained on the MNIST dataset, where features of a specific category in every sub-figure are represented in the same color: (a) features of the original teacher network; (b) features of the student network learned using the standard back-propagation strategy; (c) features of the student network learned using the proposed method with a teaching assistant.

Experiments

In this section, we implement experiments to validate the effectiveness of the proposed portable networks learning method on three benchmark datasets, including MNIST, CIFAR-10, and CIFAR-100. In addition, the experimental results are analyzed to further investigate the benefits of the proposed method.

Validations on MNIST

The MNIST dataset is a widely used dataset for conducting visual classification task with deep neural network. It is composed of 28×28 pixel grayscale digit (rom 0 to 9) images drawn from ten categories. The whole dataset of 70,000 images is split into 60,000 training and 10,000 testing images. In addition, hyper-parameters of the proposed methods in the following experiments were selected by minimizing the error on a validation set consisting of the last 10,000 training images, and optimal parameters were determined by the top performance on this set. Then, we continued training models on the full 60,000 image training set to obtain the ultimate model.

Visualization of Features. A teaching assistant was introduced in Fcn. 9, and its aim is to minimize the difference between features of student and teacher networks. The features calculated from these two networks are expected to follow the similar distribution. In order to illustrate the superiority of the proposed method, we follow the setting in (Wen et al. 2016) to extract two-dimensional features using CNNs. We first trained a LeNet++ as the teacher network, which has six convolutional layers and a fully-connected layers for extracting powerful 2D deep learning features. Numbers of filters in each convolutional layer are 32, 32, 64, 64, 128, 128, and 2, respectively. All convolution filters in the network are of size 5×5 with the stride and padding are 1 and 2, respectively. This teacher network is much deeper and wider than the conventional LeNet (LeCun et al. 1998), which achieved a 99.2% test accuracy on the MNIST dataset, and the memory usage for convolution filters of this teacher network is about $2,982KB$.

Subsequently, we initialized a thinner student network with also seven layers with half convolution filters. In practice, numbers of filters in each convolutional layer are16, 16, 32, 32, 64, 64, and 2, respectively. Then, we directly train the student network using conventional back-propagation scheme on the MNIST dataset. Unfortunately, the network accuracy is only 97.2% which is significantly lower than that of its teacher, since the student network has fewer parameters.

We then trained a new student network with the same architecture and convolution filters by exploiting the proposed method as described in Fcn. 9. λ and τ were equal to 2 and 0.5, respectively, which refer to those in the knowledge distill approach (Hinton, Vinyals, and Dean 2015). γ was set to be 1.5×10^-1, which was tuned on the validation set. The learning rate η was set to be 0.01. The accuracy of the resulting student network is 99.1% which is slightly lower than that of its teacher network but much higher than that of the student network straightforwardly learned using conventional back-propagation method. In addition, the memory usage for convolution filters of this student network is about $734KB$, which only accounts for $\frac{1}{4}$ of that of the original teacher network.

Table 1: Classification error on MNIST.

Algorithm	**#params**	**Misclass**
Student-teacher learning paradigm		
Teacher	$\sim$ 361K	0.55%
Standard back-propagation	$\sim$ 30K	1.90%
Knowledge Distillation (Hinton, Vinyals, and Dean 2015)	$\sim$ 30K	0.65%
FitNet (Romero et al. 2014)	$\sim$ 30K	0.51%
Assistant-helped learning	$\sim$ 30 K	$\mathbf{0.48}\%$
State-of-the-art-methods		
Maxout Network (Goodfellow et al. 2013)		0.45%
Network in Network (Lin, Chen, and Yan 2013)		0.47%
Deeply-Supervised Networks (Lee et al. 2015)		$\mathbf{0.39}\%$

Moreover, features (output data of the second last layer) of the above three networks were visualized in Fig. 2. It

Table 2: The performance of the proposed method on student networks with various architectures.

Networks	#layers	#params	#mult	speed-up ratio	compression ratio	FitNet	Ours
Teacher	5	$\sim$ 9M	$\sim$ 725M	$\times 1$	$\times 1$	90.21%	
Student1	11	$\sim$ 250K	$\sim$ 30M	$\times\mathbf{13.17}$	$\times\mathbf{36}$	89.03%	89.45%
Student2	11	$\sim$ 862K	$\sim$ 108M	$\times 4.56$	$\times 10.44$	91.01%	91.17%
Student3	13	$\sim$ 1.6M	$\sim$ 392M	$\times 1.40$	$\times 5.62$	91.14%	91.31%
Student4	19	$\sim$ 2.5M	$\sim$ 382M	$\times 1.58$	$\times 3.60$	91.55%	**91.68%**

is clear that features of different categories extracted using the original teacher network (Fig. 2 (a)) are separate from each other, and thus can be easily distinguished by the following softmax layer. By contrast, features extracted by the student network (Fig. 2 (b)) trained using the conventional back-propagation are distorted, thus the accuracy of this student is lower than its teacher. However, extracted features (Fig. 2 (c)) of the student network trained using the proposed method with a teaching assistant (discriminator) are similar to those of the teacher. In specific, features of the same category (points with the same color in Fig. 2) generated by teacher and student networks were located at the same area, which also demonstrates the effectiveness of the proposed approach.

Compression Results. In order to further illustrate the superiority of the proposed method, we followed the setting in (Romero et al. 2014) to train a teacher network of maxout convolutional layers as reported in (Goodfellow et al. 2013), which has 3 maxout layers and a fully-connected layer with 48-48-24-10 units, respectively. Then, we established a student network with 6 maxout layers and a fully-connected layer, but with about 8% parameters, which is as same as that in (Romero et al. 2014) for having a fair comparison. Tab. 1 reports the classification results of different networks on the MNIST dataset. Similarly, we also reported performance of students trained by only using standard back-propagation, knowledge distillation (Hinton, Vinyals, and Dean 2015), and FitNet (Romero et al. 2014), in order to illustrate the advantage of the introduced teaching assistant.

It can be found in Tab. 1 that the student network trained using the standard back-propagation scheme obtained a 1.90% misclassification error. The student network with the same architecture utilizing the knowledge distillation achieved a 0.65% error rate. The error rate of the student network trained by exploiting the FitNet approach is 0.51% which outperforms conventional back-propagation and knowledge distillation. This accuracy is slightly lower than that of the teacher network, which demonstrates that features of the teacher network contain more useful information. As for the proposed method, we added a fully connected layer after the last layer of the student network for mapping its features into the space with the same dimensionality as that of the teacher network, which is similar to that in the FitNet (Romero et al. 2014). The resulting network generated by the proposed method achieves a 0.48% misclassification error, which overcomes other teacher-student learning methods and is comparable to the state-of-the-art methods.

Validations on CIFAR-10

The above chapter demonstrates the superiority of the proposed method for learning student with a novel teaching assistant on the MNIST dataset. Here we will verify the proposed scheme on a more complex dataset, namely CIFAR-10. The dataset is composed of 32×32 pixel RGB color images belonging to ten categories. There are $50,000$ training images and $10,000$ testing images. In addition, images in the dataset were first processed using global contrast normalization (GCA) and ZCA whitening as suggested in (Goodfellow et al. 2013; Romero et al. 2014). Moreover, the last $10,000$ training images were selected as the validation set which was used for tuning the hyper-parameters of the proposed method.

Tradeoff between compression/speed-up and accuracy. Since the CIFAR-10 dataset consists of more complex images, which cannot be easily distinguished by a readily designed neural network. It is clear that, a thin and shallow student network could provide higher compression and speed-up ratios but bring an accuracy decline. Therefore, we first tested performance of several student networks for investigating the trade-off between compression performance and accuracy.

First of all, we followed the maxout convolutional network (Goodfellow et al. 2013; Romero et al. 2014) to train a teacher network composing of three convolutional layers of 96-192-192 units, respectively. A fully-connected layer of 500 units with 5-linear-piece maxout activations is then built on the top of the convolutional module. The teacher network was trained using conventional stochastic gradient descent (SGD) with learning rate decay and momentum strategies.

We then followed the experimental setting in (Romero et al. 2014) to establish four student networks, which have various architectures with different number of layers and parameters. The original teacher network has 5 convolutional layers and about $9M$ parameters. In contrast, numbers of parameters of these student networks are $250K$, $862K$, $1.6M$, and $2.5M$, respectively. The compression ratio and the speed-up ratio of each student network can be directly calculated by comparing its numbers of parameters and the floating number multiplications to those of the teacher network, respectively.

Tab. 2 reports the results of the four student networks on the CIFAR-10 dataset. It is clear that a student network with fewer parameters has a lower classification accuracy but with higher compression and speed-up ratios. In addition, although these student networks have fewer parameter than that of the teacher network, their performance is close to or surpass that of the teacher network. This confirms that

Table 3: Classification results of different networks on CIFAR-10 and CIFAR-100 datasets.

Algorithm	**#layers**	**#params**	**CIFAR-10**	**CIFAR-100**
Student-teacher learning paradigm				
Teacher	5	$\sim$ 9M	90.21%	62.78%
Student (Ours)	19	$\sim$ 2.5M	**91.68%**	**65.11%**
FitNet (Romero et al. 2014)	19	$\sim$ 2.5M	91.55%	64.89%
Knowledge Distillation (Hinton, Vinyals, and Dean 2015)	19	$\sim$ 2.5M	91.04%	63.07%
Multiple Teachers (You et al. 2017)	19	$\sim$ 2.5M	91.66%	65.06%
State-of-the-art methods				
Maxout Network (Goodfellow et al. 2013)			90.62%	61.43%
Network in Network (Lin, Chen, and Yan 2013)			91.20%	64.32%
Deeply-Supervised Networks (Lee et al. 2015)			**91.78%**	**65.43%**

the depth is more important than width for deep neural networks.

Tab. 2 also provides the results of student networks learned using FitNet (Romero et al. 2014). It is obvious that the performance of all the four student networks learned by exploiting GAN obtained higher accuracies, which suggests that more valuable information has been exploited from the teacher network by the proposed method than the conventional student-learning schemes.

Furthermore, there are a lot of works such as CNNpack (Wang et al. 2016) and deep compression (Han, Mao, and Dally 2016) focusing on compressing and speeding-up deep neural network. These approaches are complementary to the proposed method and other student-teacher learning paradigms. The portable student network learned by the proposed method has significantly fewer parameters, but it is still a regular network which can be further compressed by existing techniques such as quantization, clustering, and pruning.

Compared with state-of-art methods. After investigating the trade-off between compression performance and network accuracy, we compared the student network generated by the proposed method with its teacher and students produced by other student-teacher learning paradigms such as conventional back-propagation and knowledge distillation. Tab. 3 summarizes the results on the CIFAR-10 dataset of the proposed method and state-of-the-art methods. Note that the student network in our experiments has exactly the same architecture as that in (Romero et al. 2014) for having a fair comparison.

Since the proposed method provides a more powerful approach for constraining and optimizing features and weights of the student network, our student network outperforms others learned by the standard student-learning approaches, which demonstrates the superiority of the introduced teaching assistant. In addition, the classification accuracy of our student network is higher than that of its original teacher network, while requiring notably fewer parameters. In specific, the accuracy of the student network (Student4 in Tab. 2) is 91.68% with less than $\frac{1}{3}$ of its teacher's parameters. This student network achieves a $36\times$ compression ratio and a 13.17 speed-up ratio, which is much more flexible for real world applications on mobile devices.

Validations on CIFAR-100

Besides the CIFAR-10 dataset, we also conducted our experiments on the CIFAR-100 dataset, which has the same size and format with the CIFAR-10 dataset, *i.e.*, $60,000$ RGB color images of pixel 32×32. Since the CIFAR-100 dataset consists of 100 objects, which implies more challenge than the CIFAR-10 dataset. The accuracy of the baseline teacher network on this dataset is only about 62%, which is much lower than that on the CIFAR-10 dataset. Therefore, it is more meaningful to verify the performance of the proposed method on this dataset. In addition, images in this dataset were also processed using global contrast normalization and ZCA whitening. The teacher network and the student network have the same configuration with that of CIFAR-10 in the above chapter, and the number of units in the last softmax layer was changed to 100 according to the number of the whole categories. In addition, the dataset was augmented via random flipping as suggested in (Romero et al. 2014).

We used the fourth student network (Student4) as reported in Tab. 2 to conduct the experiment on the CIFAR-100 dataset. The classification results of the student network learned by the proposed method and state-of-the-art methods on the CIFAR-100 dataset were also reported in Tab. 3. The student network generated by the proposed method obtained a 65.11% accuracy and it is clear that the student network generated under the help of the proposed teaching assistant still outperforms networks learned using other student-teacher learning paradigms, which is more effective for learning portable deep neural networks from original heavy teacher networks. When compared to other methods, the network learned by exploiting the proposed method provides nearly the state-of-the-art performance, which is effective for learning portable deep neural networks for solving a wide range of requirements.

Conclusions

Here we examine the deep neural network compression problem for learning portable networks from original teacher models. Instead of directly transferring some useful information from the teacher to its student, we introduce a teaching assistant for helping the learning procedure. Different from conventional methods which adopt fixed measurements to evaluate the consistency between student and

teacher networks, we propose to learn an optimal transformation for a more accurate and thorough measurement. Therefore, a generative adversarial network is adopted for implementing the compression task, where the generator is exactly the student network and the discriminator acts as the teaching assistant. Experiments on several benchmark datasets show that the proposed method can produce portable neural networks with acceptable accuracy, which are superior to the state-of-the-art approaches for learning student networks.

Acknowledgements

We thank supports of NSFC 61375026 and 2015BAF15B00, and ARC Projects: FL-170100117, DE-180101438, DP-180103424, DP-140102164, LP-150100671

References

Arora, S.; Bhaskara, A.; Ge, R.; and Ma, T. 2014. Provable bounds for learning some deep representations. *ICML*.

Ba, J., and Caruana, R. 2014. Do deep nets really need to be deep? In *NIPS*.

Chen, W.; Wilson, J. T.; Tyree, S.; Weinberger, K. Q.; and Chen, Y. 2015. Compressing neural networks with the hashing trick. In *ICML*.

Courbariaux, M., and Bengio, Y. 2016. Binarynet: Training deep neural networks with weights and activations constrained to+ 1 or-1. *arXiv preprint arXiv:1602.02830*.

Denton, E. L.; Zaremba, W.; Bruna, J.; LeCun, Y.; and Fergus, R. 2014. Exploiting linear structure within convolutional networks for efficient evaluation. In *NIPS*.

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

Goodfellow, I. J.; Warde-Farley, D.; Mirza, M.; Courville, A.; and Bengio, Y. 2013. Maxout networks.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *NIPS*.

Han, S.; Pool, J.; Tran, J.; and Dally, W. 2015. Learning both weights and connections for efficient neural network. In *NIPS*.

Han, S.; Mao, H.; and Dally, W. J. 2016. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In *ICLR*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*.

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

Isola, P.; Zhu, J.-Y.; Zhou, T.; and Efros, A. A. 2016. Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*.

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

Ledig, C.; Theis, L.; Huszár, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; et al. 2016. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv preprint arXiv:1609.04802*.

Lee, C.-Y.; Xie, S.; Gallagher, P.; Zhang, Z.; and Tu, Z. 2015. Deeply-supervised nets. In *Artificial Intelligence and Statistics*.

Liang, J.; Jiang, L.; and Hauptmann, A. G. 2017. Webly-supervised learning of multimodal video detectors. In *AAAI*.

Lin, M.; Chen, Q.; and Yan, S. 2013. Network in network. *arXiv preprint arXiv:1312.4400*.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *CVPR*.

Luo, Y.; Liu, T.; Tao, D.; and Xu, C. 2014. Decomposition-based transfer distance metric learning for image classification. *IEEE TIP* 23(9):3789–3801.

Luo, Y.; Wen, Y.; Liu, T.; and Tao, D. 2017. General heterogeneous transfer distance metric learning via knowledge fragments transfer.

McClure, P., and Kriegeskorte, N. 2016. Representational distance learning for deep neural networks. *Frontiers in computational neuroscience* 10.

Mirza, M., and Osindero, S. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

Rastegari, M.; Ordonez, V.; Redmon, J.; and Farhadi, A. 2016. Xnor-net: Imagenet classification using binary convolutional neural networks. *arXiv preprint arXiv:1603.05279*.

Reed, S.; Akata, Z.; Yan, X.; Logeswaran, L.; Schiele, B.; and Lee, H. 2016. Generative adversarial text to image synthesis. In *ICML*.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*.

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

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2015. Imagenet large scale visual recognition challenge. *IJCV* 115(3):211–252.

Simonyan, K., and Zisserman, A. 2015. Very deep convolutional networks for large-scale image recognition. *ICLR*.

Wang, Y.; Xu, C.; You, S.; Tao, D.; and Xu, C. 2016. Cnnpack: Packing convolutional neural networks in the frequency domain. In *NIPS*.

Wang, C.; Wang, C.; Xu, C.; and Tao, D. 2017a. Tag disentangled generative adversarial networks for object image re-rendering. In *IJCAI*.

Wang, Y.; Xu, C.; Tao, D.; and Xu, C. 2017b. Beyond filters: Compact feature map for portable deep model. In *ICML*.

Wang, Z.; Deng, Z.; and Wang, S. 2016. Accelerating convolutional neural networks with dominant convolutional kernel and knowledge pre-regression. In *ECCV*, 533–548.

Wen, Y.; Zhang, K.; Li, Z.; and Qiao, Y. 2016. A discriminative feature learning approach for deep face recognition. In *ECCV*.

You, S.; Xu, C.; Xu, C.; and Tao, D. 2017. Learning from multiple teacher networks. In *ACM SIGKDD*.

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