Training Shallow and Thin Networks for Acceleration via Knowledge Distillation with Conditional Adversarial Networks

Zheng Xu*
University of Maryland
College Park

Yen-Chang Hsu
Georgia Institute of Technology
Atlanta

Jiawei Huang
Honda Research Institute
Mountain View

Abstract

There is an increasing interest on accelerating neural networks for real-time applications. We study the student-teacher strategy, in which a small and fast student network is trained with the auxiliary information learned from a large and accurate teacher network. We propose to use conditional adversarial networks to learn the loss function to transfer knowledge from teacher to student. The proposed method is particularly effective for relatively small student networks. Moreover, experimental results show the effect of network size when the modern networks are used as student. We empirically study the trade-off between inference time and classification accuracy, and provide suggestions on choosing a proper student network.

1. Introduction

Deep neural networks (DNNs) achieve massive success in artificial intelligence by substantially improving the state-of-the-art performance in various applications. For one of the core applications in computer vision, large-scale image classification [32], the accuracy reached by DNNs has become comparable to humans on several benchmark datasets. The recent progress towards such impressive accomplishment is largely driven by exploring deeper and wider network architectures. Despite the significant performance boost of modern DNNs [11, 46, 42], the heavy computation and memory cost of these deep and wide networks makes it difficult to directly deploy the trained networks on embedded systems for real-time applications. In the meantime, the demand for low cost networks is increasing for applications on mobile devices and autonomous cars.

Do DNNs really need to be deep and wide? Early theoretical studies suggest that shallow networks are powerful and can approximate arbitrary functions [8, 13]. More recent theoretical results show depth is indeed beneficial for the expressive capacity of networks [9, 39, 23, 33]. Moreover, the overparameterized and redundant networks, which can easily memorize and overfit the training data, surprisingly generalize well in practice [48, 2]. Various explanations have been investigated, but the secret of deep and wide networks remains an open problem.

On the other hand, empirical studies suggest that the performance of shallow networks can be improved by learning from large networks following the student-teacher strategy [4, 3, 40, 12]. In these approaches, the student networks are forced to mimic the output probability distribution of the teacher networks to transfer the knowledge embedded in the soft targets. The intuition is that the *dark knowledge* [12], which contains the relative probabilities of "incorrect" answers provided by deep and wide networks, is informative and representative. For example, we want to classify an image over the label set (dog, cat, car). Given an image of a dog, a good teacher network may mistakenly recognize it as cat with small probability, but should seldom recognize it as car; the soft target of output distribution over categories for this image, $(0.7, 0.3, 0)$, contains more information such as categorical correlation than the hard target of one-hot vector, $(1, 0, 0)$. Training is accomplished by minimizing a predetermined loss which measures similarity between student and teacher output, such as Kullback-Leibler (KL) divergence.

In previous studies, shallow and wide student networks are trained by knowledge transfer, which potentially have more parameters than the deep teacher networks [3, 40]; ensemble of networks are used as teacher, and a student network with similar architecture and capacity can be trained [12]; particularly, a small deep and thin network is trained to replace a shallow and wide network for acceleration [30], given the best teacher at that time is the shallow and wide VGGNet [36]. Since then, the design of network architecture has advanced. ResNet [11] has significantly deepened the networks by introducing residual connections, and wide residual networks (WRNs) [46] suggest widening the networks leads to better performance. It is unclear whether the dark knowledge from the state-of-the-art networks based on residual connections, which are both deep and wide, can

*zuxh@cs.umd.edu

help train a shallow and/or thin network (also with residual connections) for acceleration.

In this paper, we focus on improving the performance of a shallow and thin modern network (student) by learning from the dark knowledge of a deep and wide network (teacher). Both the student and teacher networks are convolutional neural networks (CNNs) with residual connections, and the student network is shallow and thin so that it can run much faster than the teacher network during inference. Instead of adopting the classic student-teacher strategy of forcing the output of a student network to exactly mimic the soft targets produced by a teacher network, we introduce conditional adversarial networks to transfer the dark knowledge from teacher to student. We empirically show that the loss learned by the adversarial training has the advantage over the predetermined loss in the student-teacher strategy, especially when the student network has relatively small capacity.

Our learning loss approach is inspired by the recent success of conditional adversarial networks for various image-to-image translation applications [17]. We show that the generative adversarial nets (GANs) can benefit a task that is very different from image generation. In the student-teacher strategy, GAN can help preserve the multi-modal [1] nature of the output distribution. It is not only unnecessary, but also difficult to force a student network to exactly mimic one of the soft targets (or the average/ensemble of several teacher networks), because the student has smaller capacity than the teacher. By introducing the discriminator as in GAN, the network automatically learns a good loss to transfer the correlation between classes, i.e., the dark knowledge from teacher, and also preserves the multi-modality. We summarize the motivation for our approach in Figure 1.

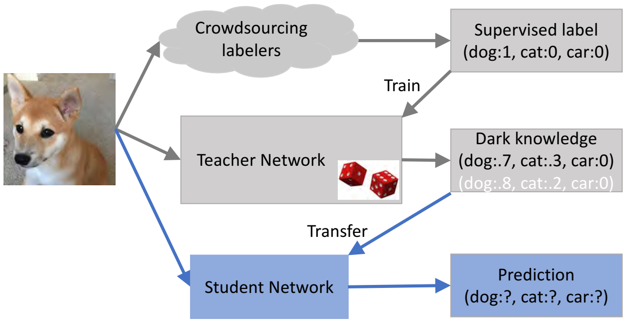

Crowdsourcing labelers
Supervised label (dog:1, cat:0, car:0)
Train
Teacher Network
Dark knowledge (dog:.7, cat:.3, car:0)
(dog:.8, cat:.2, car:0)
Transfer
Student Network
Prediction (dog:?, cat:?, car:?)

Figure 1: The motivation for our GAN-based student-teacher strategy: the soft targets produced by teacher network is more informative and the learned loss can transfer the multi-modal knowledge.

2. Related work

Network acceleration has gained increasing interest due to the growing needs of real-time applications in artificial intelligence. The techniques can be roughly divided into three categories: low precision, parameter pruning and factorization, and knowledge distillation. Low precision methods use limited number of bits to store and operate the network weights, and the extreme case is binary networks that only use 1-bit to represent each number [28, 21]. The acceleration of these methods is somewhat conceptual because mainstream GPUs only have limited support for low precision computation. Networks can also be directly modified by pruning and factorizing the redundant weights, either as a post-processing step after training, or as a fine-tuning stage [22, 14]. These methods often assume network weights are sparse or low rank, and aim to construct networks of similar architecture with reduced number of weights. Moreover, network pruning papers mostly report speedup indirectly measured in the number of basic operations, rather than by inference time directly.

Knowledge distillation is a principled approach to train small neural networks for acceleration. We slightly generalize the term *knowledge distillation* to represent all methods that train student networks by transferring knowledge from teacher networks. Bucilua *et al.* [4] pioneered this approach for model compression. Ba and Caruana[3], and Urban *et al.* [40] trained shallow but wide student by learning from a deep teacher, which were not primarily designed for acceleration. Hinton *et al.* [12] generalized the previous methods by introducing a new metric between the output distribution of teacher and student, as well as a tuning parameter. Variants of knowledge distillation has also been applied to many different tasks, such as semantic segmentation [31], pedestrian detection [35], face recognition [24], metric learning[6], reinforcement learning [38] and for regularization[34]. A recent preprint [18] presented promising preliminary results on CIFAR-10 by learning a small ResNet from a large ResNet. Another line of research focuses on transferring intermediate features instead of soft targets from teacher to student [30, 41, 47, 44, 15, 49, 45]. Our approach is complementary to those methods by directly following [12] to design a new metric between the output distribution of teacher and student, and adversarial networks are used to learn the metric to replace hand-engineering.

Generative adversarial networks (GAN) has been extensively studied over recent years since [10]. GAN trains two neural networks, the generator and the discriminator, in an adversarial learning process that alternatively updates the two networks. We apply GAN in the conditional setting [26, 17, 29, 27], where the generator is conditioned on input images. Unlike previous works that focused on image generation, we aim at learning a loss function for knowledge distillation, which requires quite different architectural

[1] We explain the multi-modality with the previous example: the output distribution for a dog image can also be (0.8, 0.2, 0). In fact, there are infinite number of soft targets that can correctly predict the label.

choices for our generator and discriminator.

3. Learning loss for knowledge distillation

In this section, we introduce the learning loss approach based on conditional adversarial networks. We start from a recap of modern network architectures (section 3.1), and then describe the dark knowledge that can be transferred from teacher to student networks (section 3.2). The GAN-based approach for learning loss is detailed in section 3.3.

3.1. Neural networks with residual connection

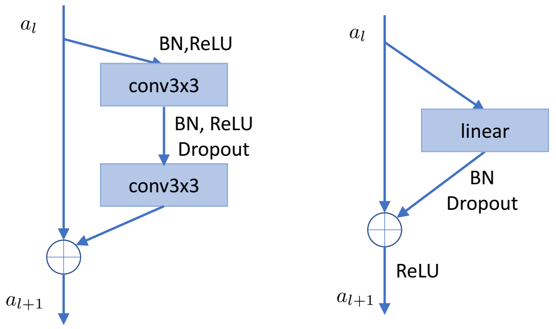


Figure 2: Residual blocks for convolutional neural networks [46] (left) and multi-layer perceptron (right). a_l represents the output of the lth block. Each block is composed of batch normalization (BN), activation ReLU, weight layer, and dropout.

The modern neural networks are built by stacking basic components. For computer vision tasks, residual blocks [11, 46] are the basic components to build deep neural networks to achieve state-of-the-art performance. Both student and teacher networks in this paper are based on the residual convolutional blocks shown in Figure 2 (left). The first layer contains 16 filters of 3×3 convolution, followed by a stack of $6n$ layers, which is 3 groups of n residual blocks, and each block contains two convolution layers equipped with batch normalization [16], ReLU [20] and dropout [37]. The output feature map is subsampled twice, and the number of filters are doubled when subsampling, as shown in Table 1. After the last residual block is the global average pooling, and then fully-connected layer and softmax. In the following sections, the architecture of wide residual networks (WRNs) is denoted as WRN-d-m following [46], where the total depth is $d = 6n + 4$, and m is the widen factor used to increase the number of filters in each residual block. Our teacher network is deep and wide WRN with large d and m, while student network is shallow and thin WRN with small d and m.

3.2. Knowledge distillation

The output of neural networks for image classification is a probability distribution over categories. The probability is generated by applying a softmax function over the output of the last fully connected layer, also known as *logits*. The dimension of logits from student and teacher networks are both equal to the number of categories. Rich information is embedded in the output of a teacher network, and we can use logits to transfer the knowledge to student network [4, 3, 40, 12]. We review the method in [12], which provides a metric between student and teacher logits that generalized previous methods for *knowledge distillation*. We denote this work as KD for simplicity.

	output size	# layers	# filters
group1	32×32	2n	16m
group2	16×16	2n	32m
group3	8×8	2n	64m

Table 1: The stacked architecture of wide residual networks [46]. n represents the number of residual blocks, m represents the widen factor.

The logits vector generated by pre-trained teacher network for an input image $x_i, i = 1, \ldots, N$ is represented by t_i, where the dimension of vector $t_i = (t_i^1, \ldots, t_i^C)$ is the number of categories C. We now consider training a student network F to generate student logits $F(x_i)$. By introducing a parameter called temperature T, the generalized softmax layer converts logits vector t_i to probability distribution q_i,

$$M_T(t_i) = q_i, \text{ where } q_i^j = \frac{\exp(t_i^j/T)}{\sum_k \exp(t_i^k/T)}. \tag{1}$$

where higher temperature T produces softer probability over categories. The regular softmax for classification is a special case of the generalized softmax with $T = 1$.

Hinton *et al.* [12] proposed to minimize the KL divergence between teacher and student output,

$$\mathcal{L}_{KD}(F, T) = \frac{1}{N} \sum_{i=1}^{N} \mathrm{KL}(M_T(t_i) \| M_T(F(x_i))). \tag{2}$$

It can be shown that when T is very large, $\mathcal{L}_{KD}$ becomes the Euclidean distance between teacher and student logits, $\|t_i - F(x_i)\|_2^2$.

When the image-label pairs $\{x_i, l_i\}$ are provided, the cross-entropy loss for supervised training of a neural network can be represented as

$$\mathcal{L}_S(F) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{H}(l_i, M_1(F(x_i))). \tag{3}$$

$\mathcal{L}_S$ is a commonly used loss for pure supervised learning in image classification from annotated data.

Finally, Hinton *et al.* [12] proposed to minimize the weighted sum of loss $\mathcal{L}_{KD}$ and loss $\mathcal{L}_S$ to train a student network,

$$\mathcal{L}_1(F,T) = \frac{1}{2}\mathcal{L}_S(F) + T^2\mathcal{L}_{KD}(F,T). \quad (4)$$

3.3. Learning loss with adversarial networks

3.3.1 Overview

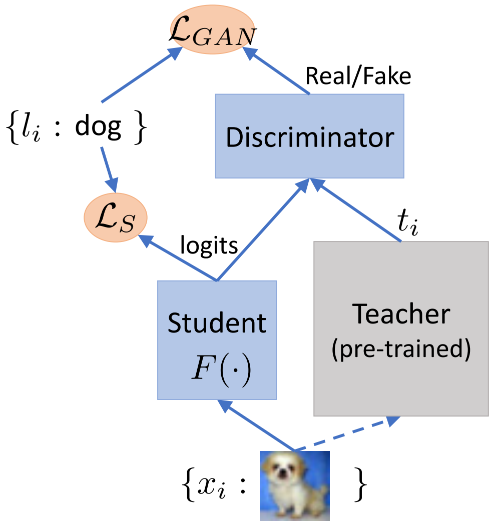


Figure 3: The GAN-based architecture to learn loss for knowledge distillation. The deep and wide teacher is pre-trained offline. The student network and discriminator are updated alternatively, where the discriminator aims to distinguish logits from student and teacher networks, and the student aims to fool the discriminator. Additional supervised loss is added for both student and discriminator.

The main idea of learning the loss for transferring knowledge from teacher to student is depicted in Figure 3. Instead of forcing the student to exactly mimic the teacher by minimizing KL-divergence in $\mathcal{L}_1(F,T)$ of Equation (4), the knowledge is transferred from teacher to student through a discriminator in our GAN-based approach. This discriminator is trained to distinguish whether the output logits is from teacher or student network, while the student (the generator) is adversarially trained to fool the discriminator, i.e., output logits similar to the teacher logits so that the discriminator can not distinguish.

There are several benefits of the proposed method. First, the learned loss can be effective, as has already been demonstrated for several image to image translation tasks [17]. Moreover, the GAN-based approach relieves the pain for hand-engineering the loss. Though the parameter tuning and hand-engineering of the loss is replaced by hand-engineering the discriminator networks in some sense, our empirical study shows that the performance is less sensitive to the discriminator architecture than the temperature parameter in knowledge distillation. The second benefit is closely related to the multi-modality of network output. Let us revisit the example of classifying a dog image from the label set (dog, cat, car). Both (0.7, 0.3, 0) and (0.8, 0.2, 0) are outputs can give correct prediction (dog), therefore it is not necessary to exactly mimic the output of one teacher network to achieve good student performance. Given the small capacity of the student network, it may not be able to exactly reproduce one particular output modality. The usage of discriminator relaxes the rigid coupling between student and teacher. The relative similarities between the categories can be captured by the discriminator trained from the multi-modal logits of teacher. Knowledge transferred from discriminator directs the student to produce output similar to the two vectors above and different from a vector like (0.5, 0.1, 0.4).

3.3.2 Discriminator update

We now describe the proposed method in a more rigorous way. The student and discriminator in Figure 3 are alternatively updated in the GAN-based approach. Let us first look at the update of the discriminator, which is trained to distinguish teacher and student logits. We use multi-layer perceptron (MLP) as discriminator. Its building block — residual block is shown in Figure 2 (right). The number of nodes in each layer is the same as the dimension of logits, i.e., the number of categories C. We denote the discriminator that predicts binary value "Real/Fake" as $D(\cdot)$. To train D, we fix the student network $F(\cdot)$ and seek to maximize the log-likelihood, which is known as binary cross-entropy loss,

$$\mathcal{L}_A(D,F) = \frac{1}{N}\sum_{i=1}^{N}\Big(\log P(\text{Real}|D(t_i)) + \log P(\text{Fake}|D(F(x_i)))\Big).$$

The plain adversarial loss $\mathcal{L}_A$ for knowledge distillation, which follows the original GAN [10], faces two major challenges. First, the adversarial training process is difficult[43]. Even if we replace the log-likelihood with advanced techniques such as Wasserstein GAN [1] or Least Squares GAN [25], the training is still slow and unstable in our experiments. Second, the discriminator captures the high-level statistics of teacher and student outputs, but the low-level alignment is missing. The student outputs $F(x_i)$ for x_i can be aligned to a completely unrelated teacher sample t_j by optimizing $\mathcal{L}_A$, which means a dog image can generate a logits vector that predicts cat. One extreme example is that the student always mispredicts dog as cat and cat as dog, but the overall output distribution may still be close to the teacher's.

To tackle these problems, we modify the discriminator objective to also predict the class labels, inspired by [5, 27]. In this case, the output of discriminator $D(\cdot)$ is a $C+2$ dimensional vector with C *Label* predictions and a *Real/Fake* prediction. We now maximize

$$\mathcal{L}_{\text{Discriminator}}(D,F) = \frac{1}{2}(\mathcal{L}_A(D,F) + \mathcal{L}_{DS}(D,F)), \quad (5)$$

where $\mathcal{L}_A$ is the previously defined adversarial loss over *Real/Fake*, $\mathcal{L}_{DS}$ is the supervised log-likelihood of discriminator over *Labels*, written as

$$\mathcal{L}_{DS}(D,F) = \frac{1}{N}\sum_{i=1}^{N}\Big(\log P(l_i|D(t_i)) + \log P(l_i|D(F(x_i)))\Big).$$

We assume *Label* and *Real/Fake* are conditionally independent in Equation (5). To avoid using this assumption, we can maximize the log-likelihood of discriminator to predict the tuple { *Label, Real/Fake* }, which requires $D(\cdot)$ to predict a $2C$ dimensional vector. In our experiments, optimizing the GAN-based loss with or without the independent assumption achieves almost identical results. Hence we will always use the independent assumption for a more compact discriminator. Note that equation (5) has the same form as the auxiliary classifier GANs [27].

The adversarial training becomes much more stable when the proposed discriminator also predicts category *Labels* besides *Real/Fake*. Moreover, the discriminator can provide category-level alignment between outputs of student and teacher. The student outputs of a dog image are more likely to learn from the teacher outputs that predict dogs.

The GAN-based loss still lacks instance-level knowledge. To exploit the knowledge to further boost the performance, we start with investigating conditional discriminators, in which the input of discriminators are logits concatenated with a conditional vector. We tried the following conditional vectors: image with convolutional embedding; label one-hot vector with embedding; and the extracted teacher logits. The embedding includes several weight layers and outputs a vector that is the same size as the logits. However, it turns out the conditional vectors are easily ignored during the training of the discriminator. The conditional discriminator does not help in practice and we introduce a more direct instance-level alignment for training student network below.

3.3.3 Student update

We update the student network after updating the discriminator in each iteration. When updating the student network $F(\cdot)$, we aim to fool the discriminator by fixing discriminator $D(\cdot)$ and minimizing the adversarial loss $\mathcal{L}_A$. In the meantime, the student network is also trained to satisfy the auxiliary classifier of discriminator $\mathcal{L}_{DS}$. Besides the category-level alignment provided by $\mathcal{L}_{DS}$, we introduce instance-level alignment between teacher and student outputs as

$$\mathcal{L}_{L_1}(F) = \frac{1}{N}\sum_{i=1}^{N}\|F(x_i) - t_i\|_1. \quad (6)$$

The L_1 norm has been found helpful in the GAN-based approach for image to image translation [17].

Finally, we combine the learned loss with the supervised loss $\mathcal{L}_S$ in (3), and minimize the following objective for the student network $F(\cdot)$,

$$\begin{aligned}\mathcal{L}_{\text{Student}}(D,F) &= \mathcal{L}_S(F) + \mathcal{L}_{L_1}(F) + \mathcal{L}_{GAN}(D,F),\\ \text{where } \mathcal{L}_{GAN}(D,F) &= \frac{1}{2}(\mathcal{L}_A(D,F) - \mathcal{L}_{DS}(D,F)).\end{aligned} \quad (7)$$

The sign of $\mathcal{L}_{DS}$ is flipped in (5) and (7) because both the discriminator and student are trained to preserve the category-level knowledge.

The final loss $\mathcal{L}_{\text{Student}}(D,F)$ in (7) is a combination of the learned loss for knowledge distillation and the supervised loss for neural network, and may look complicated at the first glance. However, each component of the loss is relatively simple. Moreover, since both student F and discriminator D are learned, there is no explicit parameters to be tuned in the loss function. Our experiments in the next section suggest the performance of the proposed method is reasonably insensitive to the discriminator architecture and the learned loss can outperform the hand-engineered loss for knowledge distillation.

4. Experiments

We present the experimental results in this section. The implementation details and experimental settings are provided in section 4.1. We show the benefits of our proposed method compared to knowledge distillation in section 4.2. We then analyze the different loss components of the proposed methods in section 4.3. The effect of depth and width of the student network is presented in section 4.4, followed by the discussion of trade-off between classification accuracy and inference time in section 4.5. Finally in section 4.6, we show the qualitative visualization on the output distribution for student, teacher, and knowledge distillation.

4.1. Experimental setting

We consider three image classification datasets: ImageNet32 [7], CIFAR-10 and CIFAR-100 [19]. ImageNet32 is a downsampled version of the ImageNet2012 challenge dataset [32], which contains 1.28M training images and 50K validation images for 1K classes; all images are downsampled to 32×32. The CIFAR datasets contain 50K training images and 10K validation images of 10 and 100 classes, respectively. The images are also 32×32. In all the experiments, we perform light data augmentation with horizontal flipping, padding and cropping on input images as in [11].

We use wide residual networks (WRNs) [46] as both student and teacher networks. The residual blocks are shown in Figure 2 (right) and the network architectures are in Table 1. WRN-d-m denotes network with depth d and widen factor m. The teacher network is a fixed WRN-40-10, while the student network has varying depth and width in different experiments. Dropout ratio of 0.3 is used for all WRNs. We use stochastic gradient descent (SGD) as optimizer, and set the initial learning rate as 0.1, momentum as 0.9, and weight decay as 1e-4. For CIFARs, we use minibatch size 128 and train for 200 epochs with learning rate divided by 10 at epoch 80 and 160. For Imagenet32, we use minibatch size 256 and train for 70 epochs with learning rate divided by 10 at epoch 25 and 50.

We use multi-layer preceptron (MLP) as the discriminator in the GAN-based approach. 3-layer MLP is used for most of the experiments except for section 4.3, in which we study the effect of discriminator depth. To speed up the experiments, the logits of teacher network are generated offline and stored in memory. For training the discriminator, we use SGD with the same scheduler as in training the student network, but a smaller initial learning rate 1e-3. The logits pass through a batch normalization layer before the MLP. Dropout ratio is also set to 0.3.

The implementation is in PyTorch. The results below are the median of five random runs.

4.2. Benefits of learning loss

We first show the proposed method is effective for transferring knowledge from teacher to student. Table 2 shows the error rate of classification on the three benchmark datasets. The teacher is the deep and wide WRN-40-10. The student is much shallower and thinner, WRN-10-4 for CIFARs, and WRN-22-4 for ImageNet32. We choose a larger student network for ImageNet32 because the dataset contains more samples and categories. Section 4.4 and 4.5 have more discussion on wisely choosing the student architecture.

	CIFAR-10	CIFAR-100	ImageNet32
Student	7.46	28.52	48.2
Teacher	4.19	20.62	38.41
KD (T=1)	7.27	28.62	49.37
KD (T=2)	7.3	28.33	49.48
KD (T=5)	7.02	27.06	49.63
KD (T=10)	6.94	27.07	51.12
Ours	**6.09**	**25.75**	**47.39**

Table 2: Error rate achieved on benchmark datasets.

The first two rows of Table 2 show the performance of pure supervised learning for student and teacher networks, without any knowledge transfer. We then compare our GAN-based approach with knowledge distillation (KD) proposed in [12] and reviewed in section 3.2. We choose the temperature parameter $T \in \{1, 2, 5, 10\}$ following the original work. The GAN-based approach is detailed in section 3.3 and no parameter is tuned.

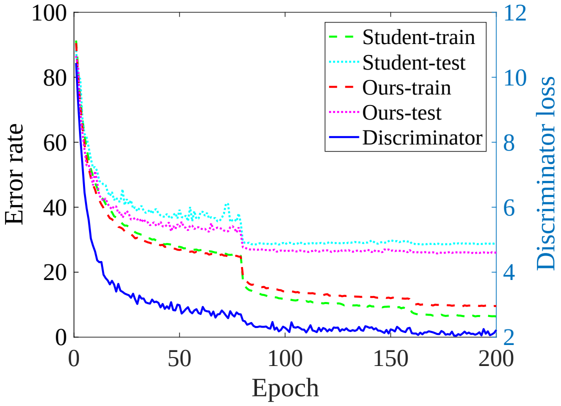


Figure 4: The training curve on CIFAR-100. We show the training and testing accuracy of the student network using supervised training and GAN-based training, as well as the discriminator loss.

We have several observations from Table 2. The deep and wide teacher performs much better than the shallow and thin student by pure supervised learning. The error rate of the small network trained with student-teacher strategy is lower bounded by the teacher performance, as expected. Baseline method KD helps the training of small networks for the two CIFARs, but does not help for ImageNet32. We conjecture the reason to be that the capacity of the student is too small to learn from knowledge distillation for larger dataset such as ImageNet32. The temperature parameter T introduced in KD is useful. For CIFARs, KD performs better when T is large, and $T = 5$ and $T = 10$ performs similarly. The proposed method improves the performance of small network for all three datasets, and outperforms KD by a margin.

4.3. Analysis of the proposed method

We discuss the proposed method in more detail in this section. Figure 4 presents the training curve of the small student network, WRN-10-4, on CIFAR-100 dataset. The loss of the discriminator (blue solid line) is gradually decreasing, which suggests the adversarial training steadily makes progress. The error rates of GAN-based method for both training and testing data are decreasing. The testing error rate of GAN-based method is consistently better than the pure supervised training of the student model, and looks more stable between epoch 50-100. Surprisingly, the training error rate of the GAN-based method is slightly worse than pure supervised learning, which suggests knowledge transfer can be more beneficial for generalization.

Loss composition	CIFAR-10	CIFAR-100
$\mathcal{L}_S$	7.46	28.52
$\mathcal{L}_{GAN}$	14.82	47.04
$\mathcal{L}_S + \mathcal{L}_{GAN}$	6.56	27.27
$\mathcal{L}_S + \mathcal{L}_{L_1}$	6.44	26.66
$\mathcal{L}_S + \mathcal{L}_{L_1} + \mathcal{L}_{GAN}$	**6.09**	**25.75**

Table 3: The effect of different components of the loss in the proposed method; the error rates on CIFARs.

Next, we look into the effect of enabling and disabling different components of the GAN-based approach, as shown in Table 3. By combining the adversarial loss and the category-level knowledge transfer (Equation (5)), the learned loss $\mathcal{L}_{GAN}$ performs reasonably well. However, the indirect knowledge provided by $\mathcal{L}_{GAN}$ alone is not as good as pure supervised learning $\mathcal{L}_S$. Both category-level knowledge transfer by $\mathcal{L}_{GAN}$ and instance-level knowledge transfer by $\mathcal{L}_{L_1}$ can improve the performance of training student network. The final approach combines these components and performs the best without parameter tuning.

Depth	1	2	3	4
Error rate	26.13	25.88	**25.75**	27.42

Table 4: The effect of discriminator depth on CIFAR-100.

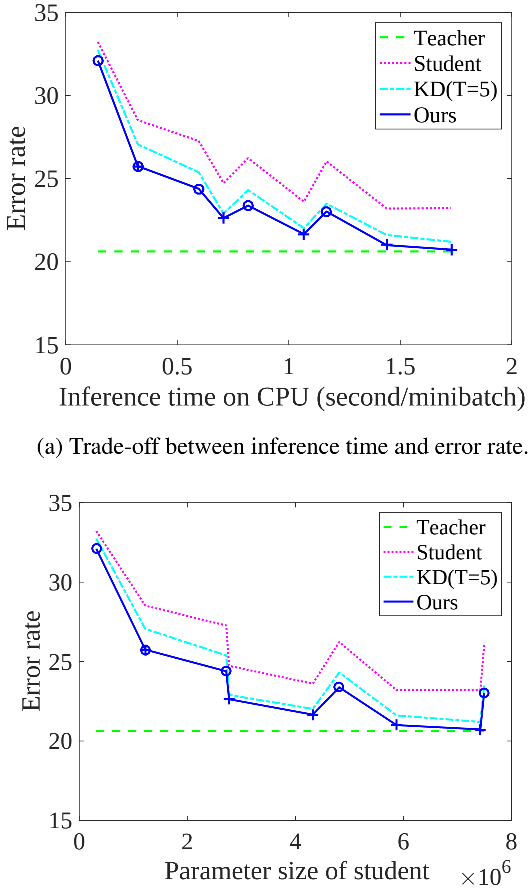
Finally, we present the effect of the depth of MLP as discriminator in Table 4. The error rate is relatively insensitive to the depth of discriminator. The error rate slightly decreases as the depth increases when the discriminator is generally shallow. When the discriminator becomes deeper, the error rate increases as the adversarial training becomes unstable. Decreasing the learning rate of discriminator sometimes helps, but it may introduce parameter tuning for the proposed method. The 3-layer MLP works reasonably well and is used for all our experiments to keep the GAN-based method simple.

4.4. Does WRN need to be deep and wide?

[40] asked similar question for convolutional neural networks and claimed the network should at least has a few layers of convolutions. We study the modern architecture WRN of residual blocks. Our empirical study suggests that even for the modern architecture WRN, the network has to be deep and wide to some extent.

Table 5 presents the results of pure supervised learning, knowledge distillation [12] and the GAN-based approach for different student networks on CIFAR-100. We first fix the depth of WRN as 10, and change the widen factor from 2 to 10. 10 is the minimum depth for our WRN architecture as the depth has to be $6n + 4$. We then fix the width as 4, and increase depth from 10 to 34. The parameter size is in millions, and the inference time is in seconds per minibatch of 100 samples on CPU.

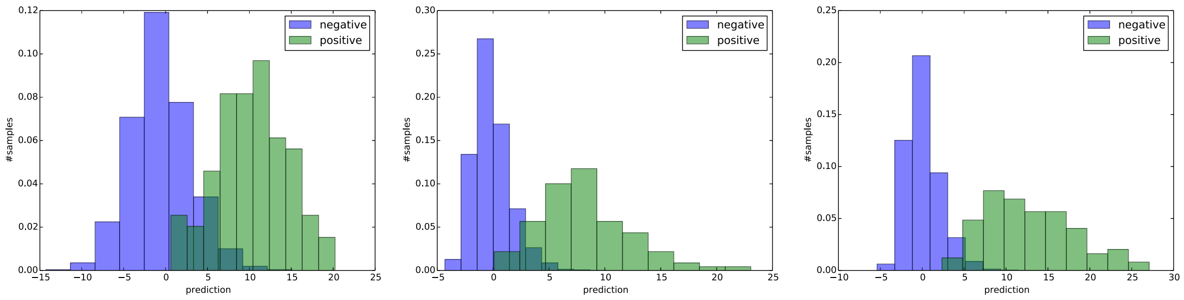
When the student is very small, such as WRN-10-2, it is difficult to transfer knowledge from teacher to student because the student is limited by its network capacity. When the student is large, such as WRN-34-4, both knowledge distillation and GAN-based approach can improve the performance close to the level of the teacher. The advantage of the proposed method is observed at all depths and widths but is most pronounced for relatively small students such as WRN-10-4. Increasing depth is more effective than increasing width for WRN. For example, WRN-34-4 has less parameter than WRN-10-10, but achieves lower error rate.



(a) Trade-off between inference time and error rate.

(b) Trade-off between network size and error rate.

Figure 5: Error rate to inference time and parameter size. The figure is generated from Table 5. Networks WRN-10-m are labeled as circles, and WRN-d-4 are labeled as crosses for the GAN-based approach. The largest student is 7x smaller and 5x faster than the teacher WRN-40-10.



(Student) (Ours) (Teacher)

Figure 6: Qualitative visualization; the distribution of prediction for category 85 in CIFAR-100.

WRN	Size (M)	Time (s)	Student	KD (T=5)	**Ours**
10-2	0.32	0.14	33.22	32.74	**32.1**
10-4	1.22	0.32	28.52	27.16	**25.75**
10-6	2.72	0.60	27.27	25.39	**24.39**
10-8	4.81	0.82	26.23	24.31	**23.38**
10-10	7.49	1.17	26.04	23.49	**23.02**
16-4	2.77	0.71	24.73	22.9	**22.73**
22-4	4.32	1.07	23.61	22.02	**21.66**
28-4	5.87	1.44	23.2	21.61	**21.00**
34-4	7.42	1.73	23.22	21.2	**20.73**
40-10	55.9	8.73	20.62	-	**-**

Table 5: The effect of depth and width in student network; the parameter size, inference time and error rate on CIFAR-100.

4.5. Training student for acceleration

The shallow and thin network is much easier to deploy in practice. We present the trade-off between error rate, inference time and parameter size in Figure 5. The figure is generated from Table 5 by changing the architecture of the student network. Larger student network is more accurate but also slower. For network with similar size, such as WRN-10-10 and WRN-34-4, deeper network achieves lower error rate, while wider network runs slightly faster. The student-teacher strategy can help improve the classification performance of the student network. When the student network is relatively large, such as WRN-34-4, the student network trained by the GAN-based approach can achieve error rate comparable to the teacher WRN-40-10, while being 7x smaller and 5x faster. Compared to the baseline student by pure supervised training, the GAN-based approach decreases the absolute error rate by 2.5%.

4.6. Visualization of distribution

In the last section of experimental results, we present qualitative visualization for the GAN-based approach. Figure 6 presents the scaled histogram for the prediction of category 85 in CIFAR-100. The histogram is calculated on the 10K testing samples, in which 100 samples are from category 85 and labeled as positive (green in figure), and the other 9.9K are labeled as negative (blue in the figure). The histogram is normalized to sum up to one for positive and negative, respectively. The three plots represent the distribution predicted by student network trained by pure supervised learning, the student network trained by GAN-based approach, and the teacher network. The histogram in the middle is similar to the histogram on the right, which suggests the GAN-based approach effectively transfers knowledge from teacher to student.

5. Conclusion and discussion

We study the student-teacher strategy for network acceleration in this paper. We propose a GAN-based approach to learn the loss for transferring knowledge from teacher to student. We show that the GAN-based approach can improve the training of student network, especially when the student network is shallow and thin. Moreover, we empirically study the effect of network capacity when adopting modern network as student and provide guidelines for wisely choosing a student to balance error rate and inference time. In specific settings, we can train a student that is 7x smaller and 5x faster than teacher without loss of accuracy.

The GAN-based approach is stable and easy to implement after applying several advanced techniques in the GAN literature. The current implementation uses the stored logtis from teacher network to save GPU memory and computation. Generating teacher logits on the fly with dropout can be more reliable for the adversarial training. At last, the GAN-based approach can be naturally extended to use ensemble of networks as teacher. The logits of multiple teacher networks can be fed into the discriminator for better performance. We will investigate these ideas for future work.

References

[1] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein GAN. *ICML*, 2017. 4

[2] D. Arpit, S. Jastrzebski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio, et al. A closer look at memorization in deep networks. *ICML*, 2017. 1

[3] J. Ba and R. Caruana. Do deep nets really need to be deep? In *NIPS*, pages 2654–2662, 2014. 1, 2, 3

[4] C. Bucilu, R. Caruana, and A. Niculescu-Mizil. Model compression. In *KDD*, pages 535–541. ACM, 2006. 1, 2, 3

[5] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *NIPS*, pages 2172–2180, 2016. 4

[6] Y. Chen, N. Wang, and Z. Zhang. Darkrank: Accelerating deep metric learning via cross sample similarities transfer. *arXiv preprint arXiv:1707.01220*, 2017. 2

[7] P. Chrabaszcz, I. Loshchilov, and F. Hutter. A downsampled variant of imagenet as an alternative to the cifar datasets. *arXiv preprint arXiv:1707.08819*, 2017. 5

[8] G. Cybenko. Approximation by superpositions of a sigmoidal function. *MCSS*, 2(4):303–314, 1989. 1

[9] R. Eldan and O. Shamir. The power of depth for feedforward neural networks. In *COLT*, pages 907–940, 2016. 1

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, 2014. 2, 4

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1, 3, 5

[12] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 1, 2, 3, 6, 7

[13] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989. 1

[14] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 2

[15] Z. Huang and N. Wang. Like what you like: Knowledge distill via neuron selectivity transfer. *arXiv preprint arXiv:1707.01219*, 2017. 2

[16] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456, 2015. 3

[17] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017. 2, 4, 5

[18] S. W. Kim and H.-E. Kim. Transferring knowledge to smaller network with class-distance loss. *ICLR Workshop*, 2017. 2

[19] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. 2009. 5

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012. 3

[21] H. Li, S. De, Z. Xu, C. Studer, H. Samet, and T. Goldstein. Training quantized nets: A deeper understanding. *arXiv preprint arXiv:1706.02379*, 2017. 2

[22] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf. Pruning filters for efficient convnets. *ICLR*, 2017. 2

[23] S. Liang and R. Srikant. Why deep neural networks for function approximation? *ICLR*, 2017. 1

[24] P. Luo, Z. Zhu, Z. Liu, X. Wang, X. Tang, et al. Face model compression by distilling knowledge from neurons. In *AAAI*, pages 3560–3566, 2016. 2

[25] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley. Least squares generative adversarial networks. *arXiv preprint ArXiv:1611.04076*, 2016. 4

[26] M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 2

[27] A. Odena, C. Olah, and J. Shlens. Conditional image synthesis with auxiliary classifier gans. *ICML*, 2017. 2, 4, 5

[28] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi. XNOR-net: Imagenet classification using binary convolutional neural networks. In *ECCV*, pages 525–542. Springer, 2016. 2

[29] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. *ICML*, 2016. 2

[30] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio. Fitnets: Hints for thin deep nets. *ICLR*, 2015. 1, 2

[31] G. Ros, S. Stent, P. F. Alcantarilla, and T. Watanabe. Training constrained deconvolutional networks for road scene semantic segmentation. *arXiv preprint arXiv:1604.01545*, 2016. 2

[32] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 1, 5

[33] I. Safran and O. Shamir. Depth-width tradeoffs in approximating natural functions with neural networks. In *ICML*, pages 2979–2987, 2017. 1

[34] B. B. Sau and V. N. Balasubramanian. Deep model compression: Distilling knowledge from noisy teachers. *arXiv preprint arXiv:1610.09650*, 2016. 2

[35] J. Shen, N. Vesdapunt, V. N. Boddeti, and K. M. Kitani. In teacher we trust: Learning compressed models for pedestrian detection. *arXiv preprint arXiv:1612.00478*, 2016. 2

[36] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[37] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 15(1):1929–1958, 2014. 3

[38] Y. W. Teh, V. Bapst, W. M. Czarnecki, J. Quan, J. Kirkpatrick, R. Hadsell, N. Heess, and R. Pascanu. Distral: Robust multitask reinforcement learning. *arXiv preprint arXiv:1707.04175*, 2017. 2

[39] M. Telgarsky. Benefits of depth in neural networks. *arXiv preprint arXiv:1602.04485*, 2016. 1

[40] G. Urban, K. J. Geras, S. E. Kahou, O. Aslan, S. Wang, R. Caruana, A. Mohamed, M. Philipose, and M. Richardson. Do deep convolutional nets really need to be deep and convolutional? *ICLR*, 2017. 1, 2, 3, 7

[41] J. Wang, Z. Wei, T. Zhang, and W. Zeng. Deeply-fused nets. *arXiv preprint arXiv:1605.07716*, 2016. 2

[42] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. *CVPR*, 2017. 1

[43] A. Yadav, S. Shah, Z. Xu, D. Jacobs, and T. Goldstein. Stabilizing adversarial nets with prediction methods. *arXiv preprint arXiv:1705.07364*, 2017. 4

[44] J. Yim, D. Joo, J. Bae, and J. Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. *CVPR*, 2017. 2

[45] S. You, C. Xu, C. Xu, and D. Tao. Learning from multiple teacher networks. In *KDD*, pages 1285–1294. ACM, 2017. 2

[46] S. Zagoruyko and N. Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 1, 3, 6

[47] S. Zagoruyko and N. Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *ICLR*, 2017. 2

[48] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. *ICLR*, 2017. 1

[49] G. Zhou, Y. Fan, R. Cui, W. Bian, X. Zhu, and K. Gai. Rocket launching: A universal and efficient framework for training well-performing light net. *arXiv preprint arXiv:1708.04106*, 2017. 2