M.S. THESIS

Layer-wise Progressive Knowledge Distillation

지식 증류를위한 다단계 교사

BY

Mohammad Amin Shabani

AUGUST 2019

DEPARTMENT OF ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

M.S. THESIS

Layer-wise Progressive Knowledge Distillation

지식 증류를위한 다단계 교사

BY

Mohammad Amin Shabani

AUGUST 2019

DEPARTMENT OF ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Layer-wise Progressive Knowledge Distillation

지식 증류를위한 다단계 교사

지도교수 이경무

이 논문을 공학석사 학위논문으로 제출함

2019년 8월

서울대학교 대학원

전기 컴퓨터 공학부

모하마드 아민 샤바 니

홍길동의 공학석사 학위 논문을 인준함

2019년 8월

위 원 장: ______________________________

부위원장: ______________________________

위　　원: ______________________________

Abstract

Knowledge Distillation (KD) is a well-known method for transferring knowledge from a teacher to a student model. In this thesis, we propose a new framework for Knowledge Distillation by introducing a Layer-wise Progressive Teacher. In this regard, we propose a method to create soft targets in different levels of complexity by obtaining the probabilities from the intermediate layers of the teacher network. Our method is specially designed for the cases that there is a large gap between the teacher and the student which makes it harder for the student to mimic the teacher. In addition, we proposed focalized teacher as a method to train a better teacher for the student. The experimental results show that our method gets significantly better results in comparison with existing knowledge distillation methods.

keywords: Knowledge Distillation, Transfer Learning, Image Classification

student number: 2017-28825

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Emerge of Deep Learning and Convolutional Neural Networks provides a huge improvement in different Computer Vision tasks. Although the early networks were very small, only a few layers, new neural networks were continuously introduced to improve the accuracy and new methods such as skip connections and regularization methods opened the doors to train much larger neural networks. Currently, there are neural networks with hundreds of layers which can improve the results of the target task. However, such a large network needs a heavy amount of resources like memory and time which is not possible to use in limited situations like mobile devices.

The above problem attracts the researcher to a new research direction for transferring the knowledge of a larger network to a smaller one which is known as Transfer Learning. Considering the larger network as the Teacher and the smaller network as the student, in this work, we are focusing on one Knowledge Distillation as one of the general methods to help the student. Transferring the knowledge from a cumbersome teacher model to a smaller student with knowledge distillation (KD) is a well-known method to improve the student accuracy. Knowledge distillation [9] is proposed to use the class probabilities predicted by the teacher model as *soft targets* to provide more information and guide the student model. Born-again neural networks [5] shows that we can also use a student with a network architecture same as the teacher in order to improve the model by guiding itself. A simple view of Knowledge Distillation method is shown in Figure 1.1. Recently, [6] proposed to use the middle layers of the teacher to transfer knowledge to the middle layers of the student in a progressive setting. Several works adopt the idea of knowledge distillation to improve different tasks such as object detection [2], model compression [3, 18], super resolution [25], and data distillation [19].

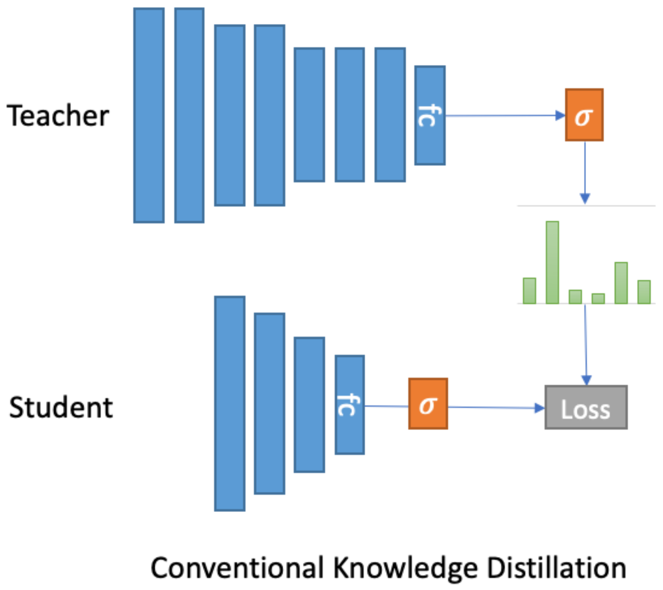


Figure 1.1: Knowledge Distillation helps a student network to mimic the soft targets provided by the teacher network.

1.2 Motivation

The accuracy of the teacher model is crucial for providing reliable soft targets, however, an ideal model can produce the final output with the probability equal to one, exactly same as the ground truth, which does not provide us any additional information to help the student model. Therefore, the differentiate between an accurate model and a good teacher is important to get the best results from the student. Considering this point, TSD method [22] uses top score difference in multiple generations to improve the quality of the soft targets. The other drawback of a strong teacher is the gap between the teacher and the student which makes it harder for the student to mimic the teacher's output. This problem is mentioned for the first time recently in [17] and the authors proposed *Teacher Assistants* which act as mediator networks to solve the problem. In this method, instead of transferring the knowledge of a teacher directly to a student, they transfer the knowledge in multiple steps with using networks of intermediate sizes.

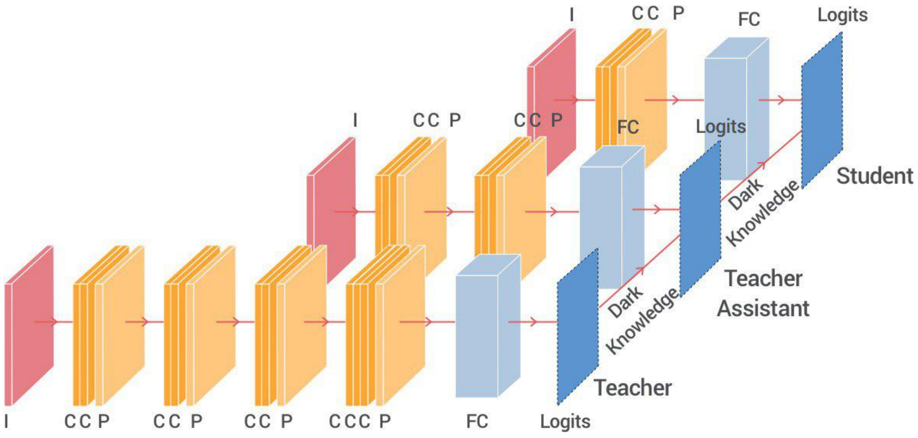


Figure 1.2: Teacher assistants are proposed to fill the gap between the teacher and the student. (This figure is taken from the original paper [17].)

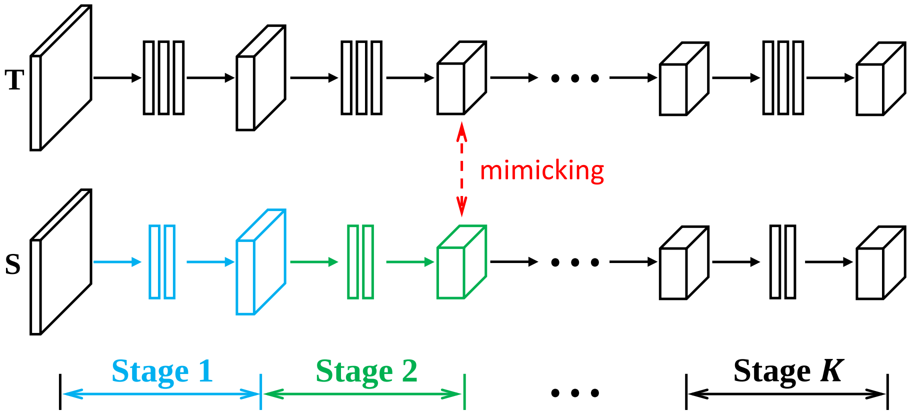


Figure 1.3: Feature Matters is also trying to use the middle layers to transfer the knowledge. (This figure is taken from the original paper [6].)

Although teacher assistants will improve the results, they need several training for obtaining each TA. In addition, the information on the teacher will be lost during each stage.

1.3 Proposed Method

In this work, first, we propose the focalized teacher in Chapter 3 to solve the high confidency problem of the teacher network by exploiting the focal loss. However, as the improvement are not enough, we propose the layer-wise progressive teacher to use the middle layers of the teacher instead of the TAs in [17]. By doing so, not only we do not need to train additional networks, we can exploit the consistency between the layers to establish a multi-level learning method to further improve the results. The intuition behind this is similar to the Curriculum Learning [1] which is a learning paradigm in that the training samples are sorted in an easy to hard order to follow the human's training process. We also do more analysis on the effect of temperature for the student and the teacher which enables us to remove the temperature of the student for simplicity of the work. Our method is also different from [6] because we transfer the knowledge of the middle layers of the teacher to the final layer of the student with a KD loss. Our method does not depend on finding appropriate layers of the student and does not have any constraints on the features sizes. Finally, we evaluate the proposed methods for Image Classification as a primary task in deep learning. For doing so, we test our methods on MNIST [14], Cifar10, and CIFAR-100 [13] in different settings. The experimental results show that the proposed multi-level method can improve the results in comparison with the other distillation methods.

1.4 Datasets

MNIST. [14] consists of a training set of 60000 examples and a test set of 10000 examples of handwritten digits with size 28×28. All of the samples are black and white images and are a subset of a larget set available from NIST.

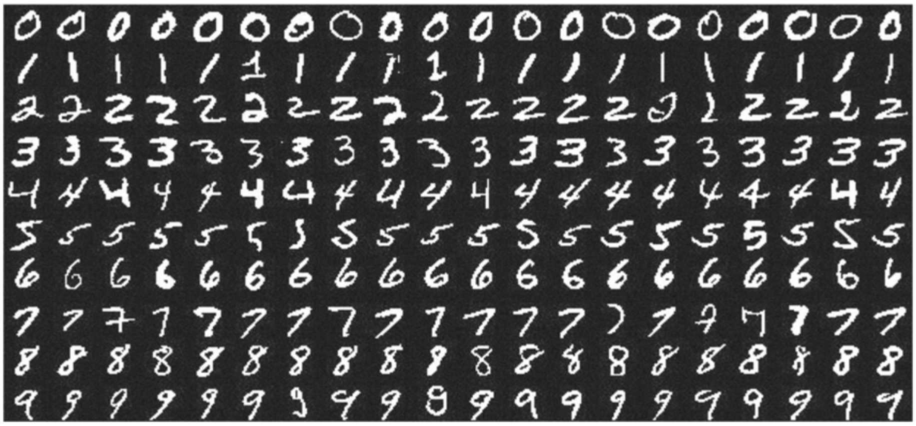


Figure 1.4: A subset of images in MNIST dataset

Cifar10 and Cifar100. [13], each of them consists of 50,000 training and 10,000 testing 32 × 32 RGB images. Cifar10 has 10 different classes airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks. Cifar100 is similar to Cifar10 except with 100 classes which are groupd in 20 superclasses such as fish, flowers, insects, trees, and people.

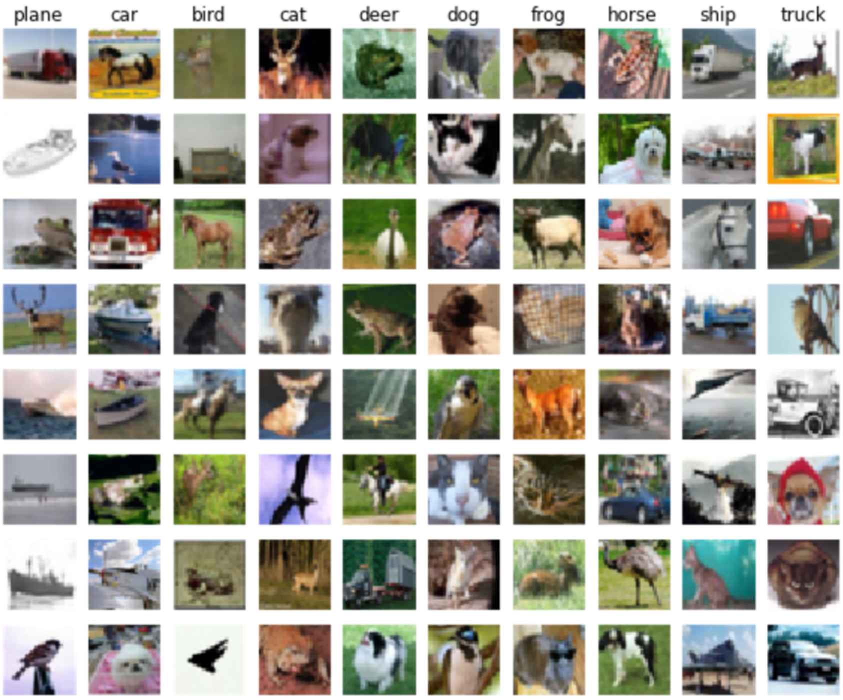


Figure 1.5: A subset of images in Cifar10 and Cifar100 datasets

Chapter 2

Related Work

2.1 Theory of Transfer Learning

There are a large number of researches attempted to transfer knowledge from a teacher model to a student model. [20] proposed FitNets, a two-stage strategy to train networks by providing *hint* from the teacher's middle layers. After that, Knowledge Distillation (KD) [9] leverage the predictions of a larger model as the *soft targets* to better training of a smaller model. This was a good start for other works to improve transferred knowledge. [5] used knowledge distillation on a student with an architecture same as the teacher to improve the performance of similar networks. They also got benefits from the embedding of several teachers with same architecture. [22] shows the importance of the secondary information for the student and proposed the top score difference which considers a specific number of semantically reasonable classes for each image as a hyperparameter. However, this parameter could be hard to estimate in more complex datasets and also it could be varied for different classes. More recently, [6] proposed to transfer the knowledge from the middle layers of the teacher to the middle layers of the student in a stage-by-stage method, Figure 1.3. This method is very similar to the first paper which is mentioned above, FitNets, but the difference is that this work uses a multi-stage method to mimic the features of the teacher. [17] mentioned the gap problem between the teacher and the student by proposing the teacher assistants, Figure 1.2. Finally, although the distilled knowledge from the softmax layer is not the only way for transferring the knowledge from a teacher to a student, it could be still helpful besides the other information. For example, [24] proposed activation-based and gradient-based attention to transfer more information from a teacher to a student and they show that knowledge distillation will still help to improve the results further.

2.2 Applications

In addition to the above literature, there are also several other works which use KD to help other tasks. In this regard, [2] improved the efficiency and the accuracy of an **object detector** by transferring the knowledge from a powerful teacher in case of the model architecture or the input data resolution to a weaker student. They also proposed Bounded Regression Loss for the bounding-box regression as a method to transfer the knowledge of a teacher for a regressions task. The proposed loss function encourages the student until it achieves the teacher's accuracy and reduces the pushing after that.

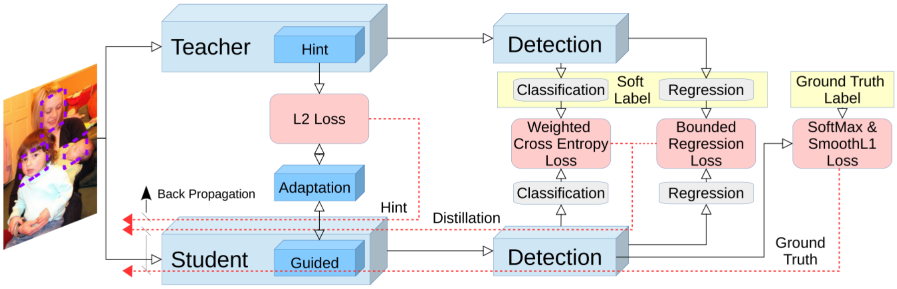


Figure 2.1: Improving object detection using knowledge distillation. The authors (This figure is taken from the original paper [2].)

In **single image super-resolution**, [25] proposed to use the knowledge of a teacher to get a good initialization for the importance of different pixels of an image. The pixels can then be used to train the student network in an easy-to-complex paradigm. In

network compression, [18] proposed quantized distillation to compress a network in terms of depth by using knowledge distillation.

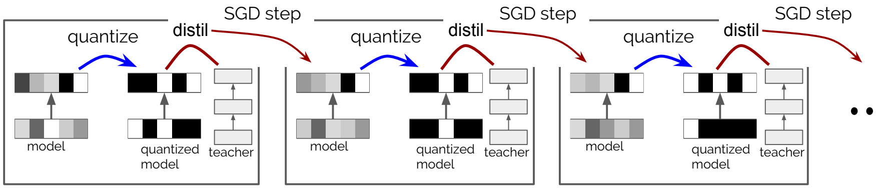


Figure 2.2: Multi-step network quantization using knowledge distillation. The authors (This figure is taken from the original paper [18].)

Finally, inspired by knowledge distillation, [19, 21] use the knowledge of a trained network for **semi-supervised learning**. [21] suggests that we can use the ensembling of the outputs from the previous iterations as teachers to predict the labels of the unlabeled images for new iterations. However, [19] makes it easier by proposing four steps data distillation to tackle omni-supervised learning. They generate annotations for unlabeled data with multiple transformations by using a trained model on a labeled dataset and then retrain the model on the union of these two datasets to improve the accuracy.

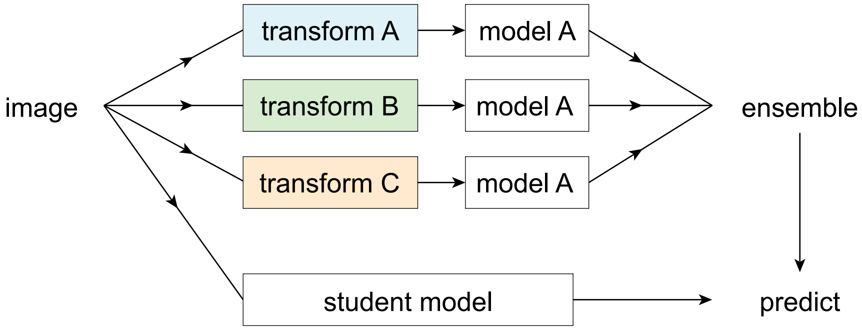


Figure 2.3: A schematic of data distillation proposed for semi-supervised learning. The authors (This figure is taken from the original paper [19].)

Chapter 3

Focalized Teacher

3.1 Overview

In this chapter, we propose a simple method by using the weighted outputs for the final loss of the teacher. The idea is based on the focal loss [15] which is first proposed to solve the class imbalance problem between the foreground and the background in dense object detection task. Focal loss decreases the weights of easy examples and focuses on the harder ones in which the hardness of an example is determined by the output certainty of the network for that example. We noticed the side effect of this procedure is that as the network focuses on the harder examples, the force on the easy examples to have probabilities equal to one will decrease in comparison with the normal classification loss functions. We benefit from this effect to produce a teacher with milder supervision signal. In addition to the mentioned point, the other problem in KD is that the predicted classes of a teacher network are not necessarily correct even in the training data. In the case of semi-supervised learning, [19] mentioned this problem by proposing to create batch sets consist of both the original and the new labeled images to have true labels in every iteration. In the case of fully supervised learning, we show that we can swap the probability of the true class with the highest probability of the predicted soft target to generate a completely valid training set.

3.2 Label Correction

We should notice that the produced labels of the teacher network do not always predict the correct class for the images. Especially, in the case of using the soft targets as the only labels for the training, it leads to an upper bound for the student accuracy based on the teacher accuracy. This forces us to provide a strong teacher who needs more effort to manage. For removing this upper bound, [9] proposed to use an additional loss function which is the cross-entropy with the one-hot vectors of the correct labels which needs to determine the weight of each loss. However, we show that we can easily switch the logits of the largest probability and the probability of the correct label of an image without harming the general distribution which is provided by the soft targets. By doing so, all of the predicted classes are true and the teacher accuracy is equal to 100 percent. Figure 3.1 shows some of the samples with false predictions in which the label correction can help to have better soft targets.

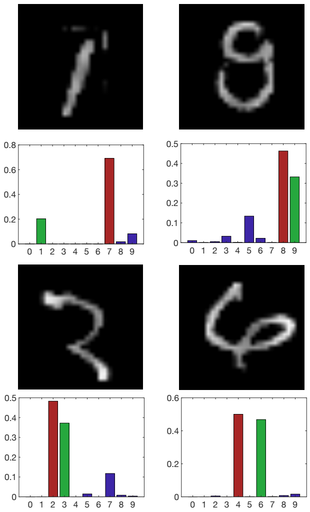


Figure 3.1: Some of the output probabilities of the teacher with wrong predicted classes for MNIST. The green color shows the ground truth class. The x-axis is the classes and the y-axis is the predicted probabilities. We also use $T = 2$ in here for better visualization of the probabilities.

3.3 Focalized Teacher

The other issue in the common knowledge distillation framework is that the teacher network does not have any constraint to produce better labels for the student and the only goal is to train a more accurate network as much as possible. Therefore, training of the teacher with a regular classification loss function, like cross-entropy, could produce very high confidence outputs for the true class which makes the probability of the other classes very small. Although increasing the temperature can provide larger probabilities, these probabilities come from very small numbers which contain a higher amount of noise. To overcome this problem, we propose using a weighing method to make the teacher less sensitive to the high confidence examples. We use the focal loss [15] which is first proposed to solve the class imbalance problem in object detection. However, we show that we can also use it to produce better soft targets. More formally, we use $(1 - p)$ as a weighing factor for the cross-entropy loss as it is shown in the following equation.

$$L(p) = -(1 - p)^{\gamma} log(p) \tag{3.1}$$

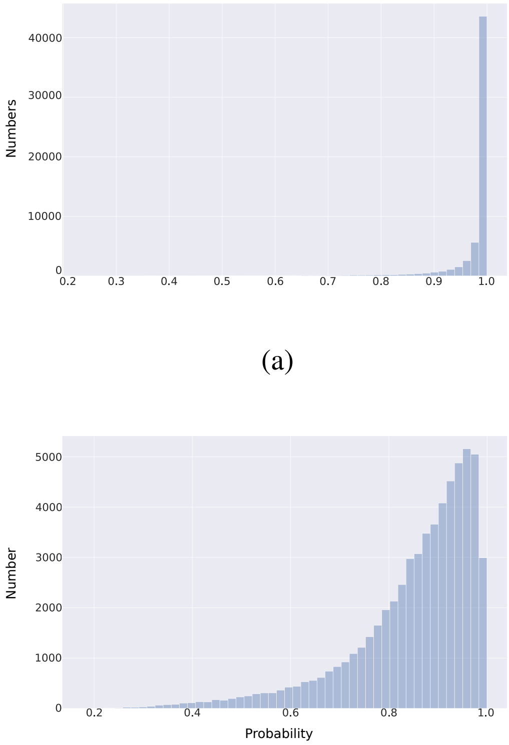
In the above equation, p is the probability of the true class which is predicted with the network and γ is the focusing parameter as in [15]. The normal cross-entropy can be obtained by setting $\gamma = 1$ and increasing the gamma causes more focussing on the harder examples. Intuitively, with using this weighing, an image with an output probability of 0.99 have almost 100 times smaller loss than an image with an output probability of 0.9. By doing this, we create a delay for the network to focus on the producing high confidence outputs as far as harder examples exist. Figure 3.2 shows the effect of this weighing to produce softer targets.

Dataset	with weighing	without weighing
CIFAR-10	71.68	70.08
MNIST	98.97	99.04

Table 3.1: The accuracy of the teachers with and without weighing.

3.4 Experimental Results

This section describes the details of the experimental results of our method. We use MNIST [14] and CIFAR-10 [13] to evaluate each part. For each dataset, the teacher networks are trained with all of the training samples to get the best results and the student networks are trained with only 500 images in CIFAR-10 and 1000 images in MNIST to emphasize on the effect of the distilled knowledge. For the teacher network, our model consists of two convolutional layers with the kernel size of 3×3, followed by two fully connected layers of size 128 and 10 for the final softmax. We also used a max-pooling layer after each convolutional layer. The student architecture is the same as the teacher except for the size of the first fully connected which is 64. For the first experiment, we show the effect of weighing on the soft targets. Therefore, we use focal loss for the weighing function and following [15], γ is set equal to 2 in all of the experiments. The comparison of the teacher's accuracy on the test data is shown in Table 3.1. In the MNIST dataset, most of the images are very easy for the network and focusing on the harder ones will help the network for getting better accuracy. However, Cifar-10 is much harder and most of the images contains useful information to train the teacher network. On the training set, the teacher with and without weighing respectively achieved 72.03 and 74.37 on Cifar-10 and 99.15 and 99.23 on MNIST. Figure 3.2 shows the effect of the weighing on the output probabilities of the training set of the MNIST. We use the above teachers to improve student accuracy. Due to the point that the ideal temperature of the teacher without weighing could be higher than the other one, for each experiment, we trained the student with temperatures from 2 to 6 and get the best



(a)

(b)

Figure 3.2: The true class output probabilities of the teachers for MNIST training set (a) with and (b) without the weighing part. For better representation of the differences, we use T equal to 2 in here.

temperature for each one for a better comparison. We also repeat each experiment four times and report the median of them. The results are shown in Table 3.2.

The baselines are trained with a simple cross-entropy loss. As you can see, the effect of the label correction in CIFAR-10 is about half of the improvement which is higher than the MNIST. It's due to the high accuracy of the teachers in MNIST. Also weighing is more helpful in the MNIST dataset because of the high number of easy examples. However, both parts improve the results in both datasets which demonstrates the effectiveness of the proposed method. For more analysis, we calculate the variance

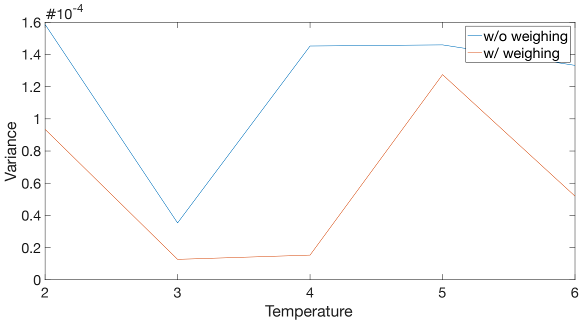


Figure 3.3: The variance of the results for CIFAR-10 within each temperature. The results of the trained models with soft targets of the proposed teacher have smaller variance in all temperatures.

Dataset	Baseline	KD [9]	WKD	WKDC
CIFAR-10	39.66	44.68	44.75	**44.81**
MNIST	90.98	92.36	92.81	**92.96**

Table 3.2: Comparison of the accuracy of the students. *KD* mean knowledge distillation, *W* means with weighing, and *C* means with the label correction part.

of the experimental results for each temperature which are shown in Figure 3.3. The figure shows that the results of the proposed method have less variance which can be interpreted as less noisiness in the classes with small values in the soft targets.

Chapter 4

Layer-wise Progressive Knowledge Distillation

4.1 Background and Notations

Knowledge Distillation [9] is proposed to transfer the knowledge of a trained teacher network to a student network. The teacher acts as a function to convert the given one-hot ground truth labels or *hard targets* of a dataset to probability distributions named as *soft targets*. Soft targets can help the student network by reducing the variance in the gradients of the training and also providing more information to the network. Formally, consider y as the ground truth label, z as the logits of the student, and v as the logits of the teacher, following [9], the knowledge distillation loss L_{KD} is as follows:

$$L_{KD} = H(\sigma(v/T), \sigma(z/T)), \tag{4.1}$$

where σ is a softmax function, T is temperature, and H is a cross entropy function. $\sigma(v/T)$ is the *soft target* in contrast with the ground truth labels which we call *hard targets*. The loss function L_{KD} is equal to the conventional cross entropy when $T = 1$, and it will encourage the student to pay more attention to the smaller probabilities as T increases.

In general, in addition to L_{KD}, a cross-entropy loss with the ground truth labels is also used to modify the soft targets [9, 17, 26]. Therefore, the total loss for the student

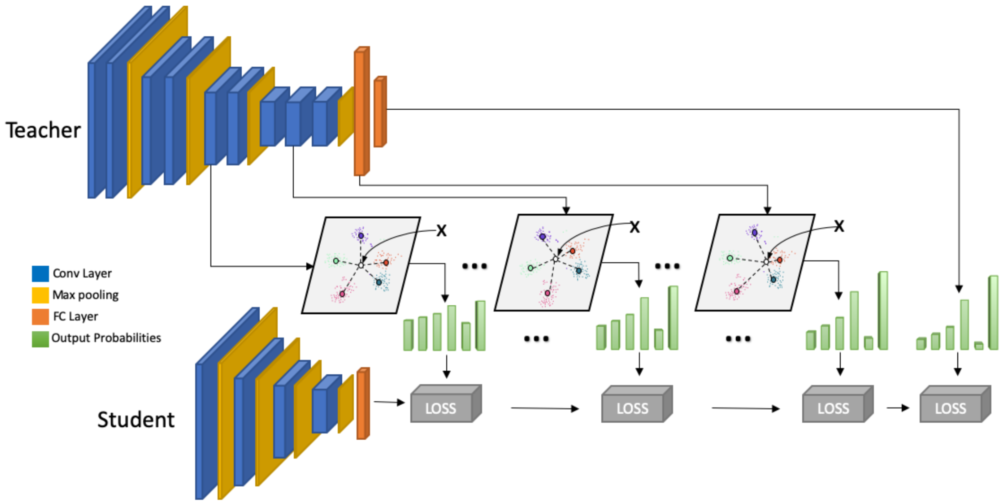


Figure 4.1: The proposed method provides easier soft targets by converting the feature spaces of the middle layers to probabilities. Therefore, instead of using just the final outputs of the teacher similar to the conventional methods on knowledge distillation, we can manage a multi-level learning for the student network.

is as follows.

$$L_{total} = (1 - \lambda)H(y, z) + \lambda T^2 \times L_{KD}, \tag{4.2}$$

where T^2 is multiplied to balance the relative contribution of the loss functions as much as possible [9].

4.2 Layer-wise Knowledge Distillation

To the best of our knowledge, all the works on knowledge distillation until now use only the softmax outputs of the final layer of the teacher as the soft targets to guide the student network. However, these probabilities obtained from the highest level of features at the end of the network could be very hard to learn for the student as demonstrated in [17]. In this section, we show how we can avoid the hardness of the final outputs of the teacher by using the hidden information of the intermediate layers.

Let x and c_j denote training data and the j-th class, respectively. Now, for a teacher network with the embedding function $f_l(.)$ of the l-th layer, to obtain the logits, we first create the class prototypes p_{lj} by

$$p_{lj} = \frac{1}{|c_j|} \sum_{x \in c_j} f_l(x), \tag{4.3}$$

where $|c_j|$ represents the cardinality of c_j. Then, the logits v_l of the l-th layer of the teacher with respect to an training sample x can be determined by the cosine similarity between the corresponding feature vector $f_l(x)$ and each class prototype such that

$$v_l : v_l(j) = \frac{f_l(x) \cdot p_{lj}}{||f_l(x)||\;||p_{lj}||}. \tag{4.4}$$

The reason why we choose the cosine similarity for calculating the logits is that it empirically gives the best results than others, including the Euclidean distance (more details in 4.4.2).

Finally, the soft target is obtained by applying softmax function over the produced logits v_l. Note that since the distribution of the intermediate logits that are resulted in by the cosine similarity in Eq. (4.4), it is quite different from those of the final outputs of the teacher and the student as well. We observe that v_l produces very smooth probabilities with small variances among different classes. Therefore, to match the distributions of intermediate soft target and the student output for proper knowledge distillation, it is crucial to differentiate the teacher's temperature and the student's temperature, when it comes to the intermediate layers. In this regard, we separate the temperature T in Eq. (4.1) into two different temperatures T_t and T_s for the teacher and the student, respectively. So, we have new layer-wise knowledge distillation loss as follows:

$$L'_{KD} = H(\sigma(v_l/T_t), \sigma(z/T_s)). \tag{4.5}$$

Note that T_t determines the entropy of the soft targets, while T_s only controls the sharpness of the final output distribution. Therefore, as we show in the later sections, we expect that T_t plays the principal role in the final performance of the distilled model.

4.3 Progressive Teacher

Inspired by humans who can learn different levels of understanding from observation of a phenomenon, the soft targets of different layers could be also considered as different levels of supervision. This concept is also similar to the Curriculum Learning [1] in which the network starts training from easier samples and considers harder samples gradually. However, instead of adding harder samples, the samples are fixed in our case and we just improve the accuracy of the labels.

For the layer-wise progressive knowledge distillation, we first select a set of m layers of the teacher. Then, we get the logits of those layers and use them to construct a progressive teacher. These logits can be obtained by Eq. (4.4) for the intermediate layers or directly from the last layer of the network.

Now, we train the network with the following loss function:

$$L_p = L'_{KD}(v_{\lceil \alpha \times m \rceil}, z) = H(\sigma(v_{\lceil \alpha \times m \rceil}/T_t), \sigma(z/T_s)), \tag{4.6}$$

where, $\lceil \cdot \rceil$ denotes ceiling function, and α is the age of the model that is the ratio of the current epoch to the total number of epochs for the training, which is between 0 and 1. Therefore, knowledge distillation starts from layer 1 and moves to higher layers progressively as training goes on. We also remove the cross-entropy loss of the ground truth which is in Eq. (4.2) for preventing from any constraint on the student to follow the teacher. Instead, as Algorithm 1 shows, we can train the student network on the ground truth labels after learning from all the teacher's layers as an additional step.

Algorithm 1: Layer-wise Progressive Knowledge Distillation (LPKD)

```
Data: A set of $m$ layers of the teacher, mapping functions $f_l(.)$ to convert from
      the image space to the feature space of $l$th layer, and the student network
      $\mathcal{S}$.
for $1 \leq l \leq m$ do
    if $l$ not equal to the final layer then
        Compute the class prototypes with Eq. (4.3);
        Obtain the logits $v_l$ for the training images $x$ and the ground truth labels
          $y$ based on the cosine similarity using Eq. (4.4);
    else
        Obtain the logits $v_l$ directly from the final output of the network.
    end
    Update the student network $\mathcal{S}$ using the logits $v_l$ and the outputs of the
      student network $z$ with $L'_{KD}(v_l, z)$ as the loss function;
end
Update the student network $\mathcal{S}$ using the labels $y$ with cross entropy loss function;
```

4.4 Experimental Results

In this section, we first introduce the experimental settings, such as the teacher, the student models, and the datasets. In Section 4.4.1, we discuss the effect of the temperatures in Eq. (4.5). After that, we show the results of knowledge distillation based on a single intermediate layer in Section 4.4.3, the progressive method in Section 4.4.4. Finally, we compare our method with the existing knowledge distillation methods in Section 4.4.5.

Datasets. For the evaluation of our method, we followed the experimental settings of [17]. We perform a set of experiments on image classification with two standard datasets CIFAR-10 and CIFAR-100 [13], each of them consists of 50,000 training and 10,000 testing 32 × 32 RGB images. As a preprocessing step on both CIFAR-10 and

CIFAR-100, we transformed images into ones with zero means. We also used horizontal flipping for the data augmentation.

Training. For the implementation, we used Keras [4] framework and we used Adam optimizer [12] for 200 epochs. The learning rate is selected between 0.001 and 0.01 based on the model and decreased after 80,120, 160, and 180 by factors of 0.5, 0.1, 0.05, and 0.001. The precision of hyperparameter tuner is 5, 1/5, and 0.1 for the temperatures of the final layers, temperatures of the intermediate layers, and the lambdas, respectively. We used the Neural Network Intelligence optimization toolkit [16] to find the best parameters for each experiment. The reported results are the maximum among three trainings with different random seeds.

Networks. As in [17], we used two types of networks named by Plain and ResNet [7] with different sizes. The Plain CNN consists of simple convolutional layers followed by batch normalization [10] and ReLU activation. We use two networks for the Plain type; Plain2, the student model that is composed of just 2 convolutional layers, a max-pooling after each one and a fully connected layer at the end, and Plain10, the teacher model that has 10 convolutional layers and two fully connected layers at the end. The max-poolings are after the 2nd, 4th, 6th, and the 10th layers. These two networks are shown in Figure 4.1. For the test of more complex networks, we used the original structure of ResNet with 8 blocks as the student and ResNet with 110 blocks as the teacher. We also use Resnet26 and Resnet14 as the teacher and the student, respectively, to compare our method with the others. For all experiments on the intermediate layers, we got the output of the layer after the activation function.

4.4.1 Temperature Analysis

Following Section 4.2, we first analyze the effect of each temperature T_t and T_s on the final accuracy of the distilled model. For doing so, we trained a teacher with architecture Plain10 for the plain version and Resnet110 for the Resnet version. We used the final outputs of these teachers and L'_{KD} in Eq. (4.5) to train Plain2 and Resnet8 with 4

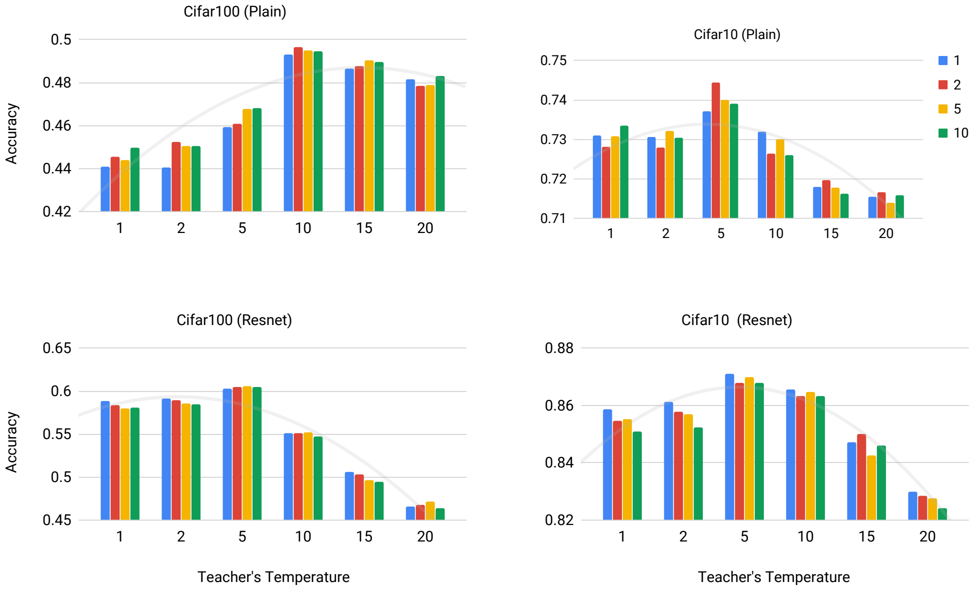


Figure 4.2: The accuracy of the student model by using different temperatures for the teacher and the student. The values in the legends and the X-axis indicate students' temperatures and the teachers' temperatures, respectively. Gray lines are fitted polynomial functions of degree 2 and show the trends.

different temperatures from 1 to 10 for T_s, and 6 different temperatures from 1 to 20 for T_t.

Figure 4.2 shows the test results. In all experiments, we observe that the teacher's temperature T_t has the principal effect on the knowledge distillation performance. It is because the teacher's temperature will directly affect the soft targets and controls the attention of the student to match smaller logits. If the soft target's distribution is fixed, however, the student network can adapt to that distribution during the training procedure based on the value of T_s. This makes T_s only responsible for determining the student's output distribution after training. Based on these results, we empirically fixed T_s to be 1 for a smaller search space of hyperparameters, without sacrificing performance.

4.4.2 Distance Metric

In this part, we compare the results of the intermediate layers with three metrics; Cosine Similarity, Euclidean distance, and Correlation. For each one, we follow a same process like Eq. (4.3) and Eq. (4.4) to obtain the logits and we consider the maximum value among the logits as the predicted class.

Table 4.1: Comparison of the accuracy of different layers of the Plain10 based on the selected metric on CIFAR-10.

Metric	**Conv7**	**Conv8**	**Conv9**	**Conv10**	**FC1**
Cosine	84.24	**87.22**	87.28	**88.19**	**88.23**
Euclidean	80.87	82.1	84.93	84.87	87.32
Correlation	**84.32**	**87.22**	**87.33**	88.11	88.19

Table 4.2: Comparison of the accuracy of different layers of the Plain10 based on the selected metric on CIFAR-100.

Metric	**Conv7**	**Conv8**	**Conv9**	**Conv10**	**FC1**
Cosine	**51.01**	**53.53**	55.39	**55.89**	**59.98**
Euclidean	50.34	52.88	54.88	52.97	57.9
Correlation	50.99	53.47	**55.81**	55.62	59.56

The results are shown in Table 4.1 for CIFAR-10 and Table 4.2 for CIFAR-100. For both datasets, the Euclidean distance got the worst results and comparing the Correlation and Cosine similarity we can conclude that subtracting the means after activations is not necessarily required. Therefore, we just using the Cosine similarity in this work.

4.4.3 Distilled Knowledge from an intermediate layer

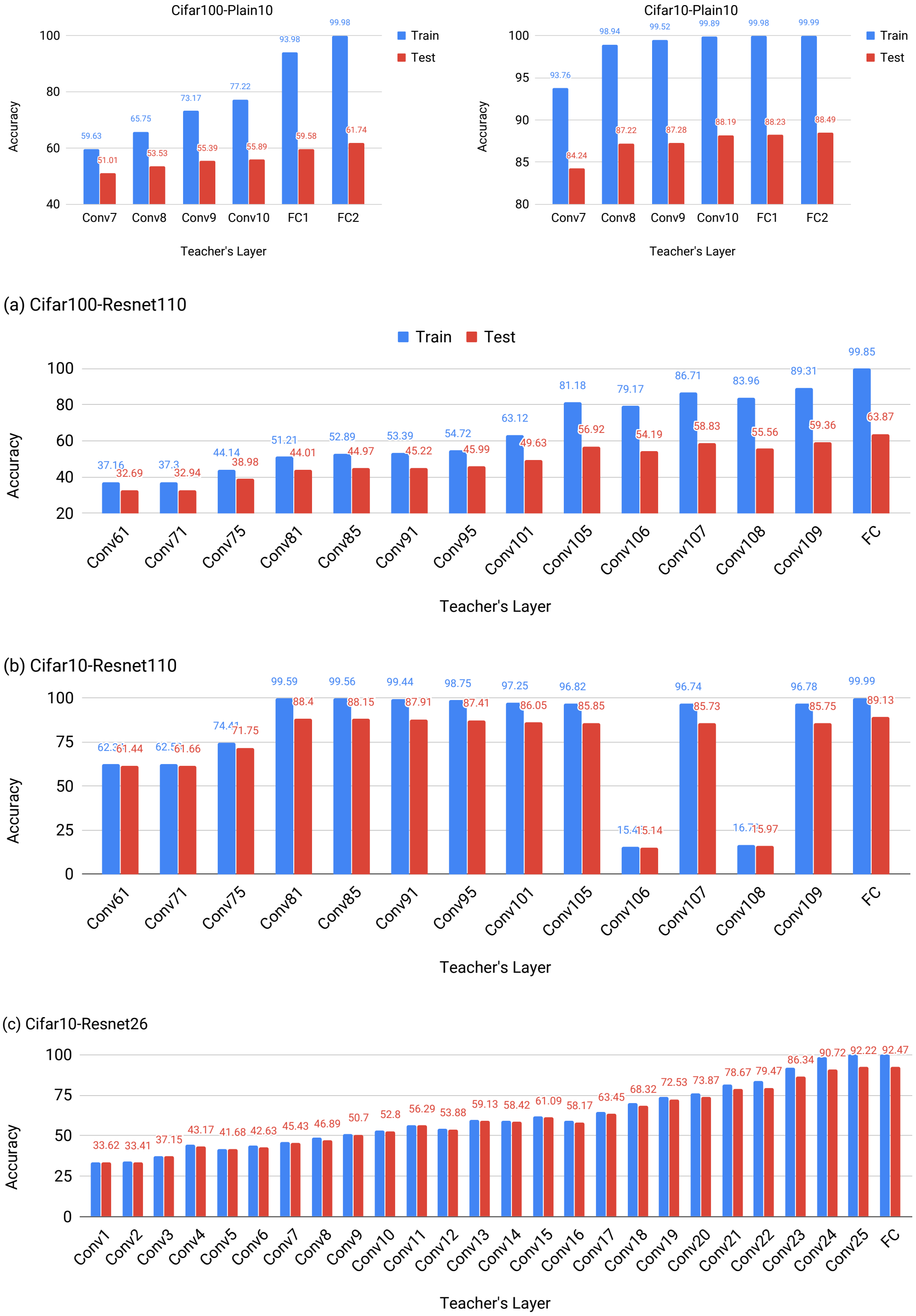
To validate and compare the quality of the transferred knowledge from each layer of the teacher, we used the last 4 convolutional layers of the Plain10 in addition to the two fully connected layers at the end of the network. The training and the test accuracy of each layer of Plain10 and the ResNet models are shown in Figure 4.3. Considering the accuracy of the teacher's layers is important especially in the case of more complex networks like Resnet in which some of the middle layers could contain very limited information due to the skip connections. However, there should be a flow of the information during the layers of the network which give us a subset of layers with increasing accuracy.

In order to train the student network Plain2, we used the normal output probabilities and Eq. (4.2) for the final FC layer. For the intermediate layers, we used feature to logits of Section 4.2 and the Eq. (4.5) as the loss function. The results are shown in Table 4.3.

Table 4.3: Comparison of the knowledge distillation performance with using different layers of the teacher.

Dataset	**NOKD**	**Conv7**	**Conv8**	**Conv9**	**Conv10**	**FC1**	**FC2(KD)**
CIFAR-100	44.39	46.2	47.68	48.61	48.37	49.68	49.54
CIFAR-10	73.00	73.99	74.12	74.62	74.3	74.13	74.06

Not only the intermediate layers can guide the student properly, they even can surpass the result which comes by guiding of the final layer. In addition, considering the accuracy of the teacher's layers in CIFAR-100, the teacher's accuracy gradually improved specifically in layers Conv8, Conv9, FC1, and FC2 which is reflected in the student's accuracy until layer FC1. The result of the student for layer FC2 is lower which could be due to the trade-off between the accuracy and the complexity of the teacher. The results on CIFAR-10 is also similar but the layers' accuracies are saturated in earlier layers.



(a) Cifar100-Resnet110

(b) Cifar10-Resnet110

(c) Cifar10-Resnet26

Figure 4.3: The train and the test accuracy from the different layers of Plain10 and ResNet models.

We note that the distributions that are obtained from the cosine similarity are very smooth with small probabilities. Figure 4.4 shows the histogram of the output probabilities for the predicted class of the validation data on CIFAR-100 and the Plain10 model. These are very different from the distribution of the final layer which comes directly from the network's output.

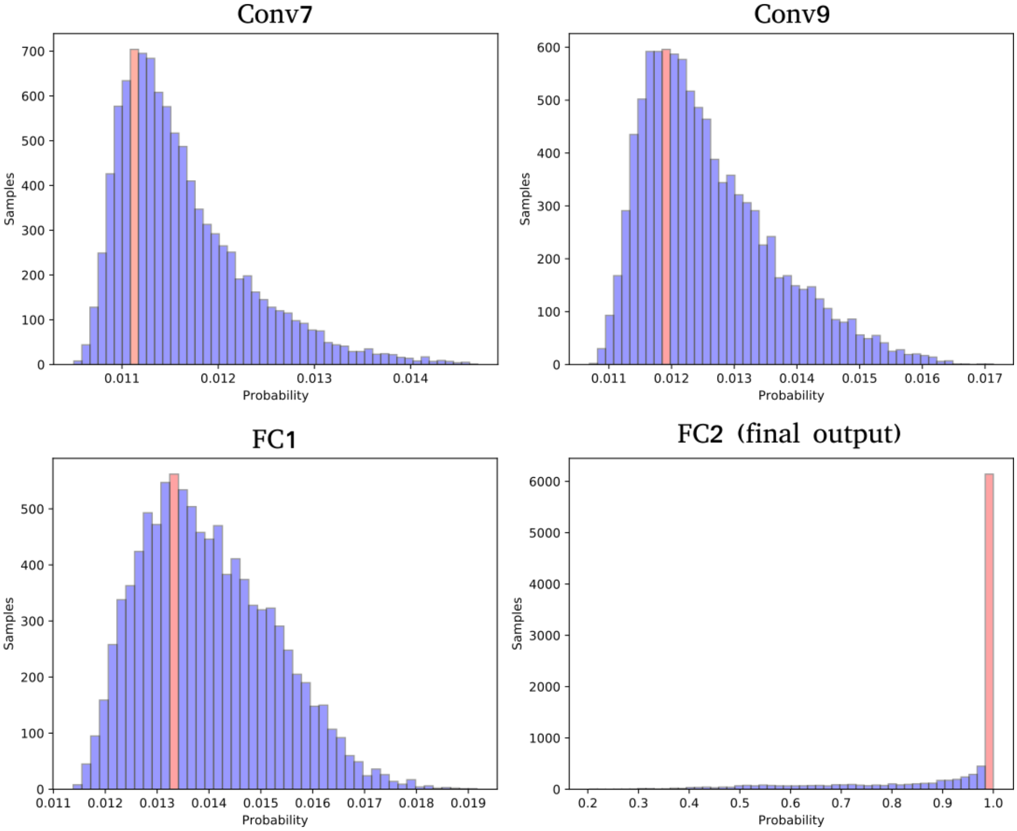


Figure 4.4: Histogram of the highest probabilities for each sample for different layers of Plain10. The pink line indicates the probability with the maximum number of samples in each layer.

Considering only the intermediate layers (Conv7, Conv8, and FC1), the position of the pink line, which means the probability that highest number of samples are predicted the output with this probability, is moved from around 0.011 to 0.013 which demonstrates sharper distributions in higher layers.

4.4.4 Progressive Teacher

Training in a progressive way can help the student by preparing the network for harder labels in each step. In this regard, the training of the proposed method consists of three parts according to Algorithm 1; in the first part, we use the intermediate layers in a selected set of the layers to train the student network with the same temperature for all of them, the second part uses the soft targets from the final layer with a new temperature, and the third part is cross-entropy with the original labels. Each training part follows the same setting in the case of epochs and we found the best learning rate for each part.

Following the above settings, we first trained Plain2 (Resnet8) model as the student with the soft targets of only one intermediate layer and the final layer of the teacher, Plain10 (Resnet110).

Table 4.4: The accuracy of Plain2 learned by different intermediate layers in addition to the final outputs of Plain10.

Dataset	**Conv7**	**Conv8**	**Conv9**	**Conv10**	**FC1**
CIFAR-100	52.6	52.86	52.75	53.23	52.81
CIFAR-10	75.22	74.83	74.54	74.52	74.47

Table 4.5: The accuracy of Resnet8 learned by different intermediate layers in addition to the final outputs of Resnet110.

Dataset	**Conv105**	**Conv106**	**Conv107**	**Conv108**	**Conv109**
CIFAR-100	60.89.6	60.72	61.29	61.24	60.95
	Conv79	**Conv81**	**Conv105**	**Conv107**	**Conv109**
CIFAR-10	87.39	87.79	87.5	87.34	87.44

According to the results which are shown in Table 4.4 and Table 4.5, the best accuracy is achieved from the intermediate layers which are almost in the middle of the final layer of the teacher and the KD version of the student in terms of accuracy. This is similar to the results of finding the best teacher assistant in [17].

We also provide the results of the proposed layer-wise progressive knowledge distillation method with more than one intermediate layer in Table 4.6. For the baselines, we trained the networks with the cross-entropy loss of the ground truth labels, KD is the networks which are trained with the loss of both soft targets of the final layers and the ground truth labels L_{total}, described in Eq. (4.2). LPKD is our progressive method with the soft targets of the intermediate layers in addition to the soft targets of the final layer and the ground truth labels at the end. For the Plain version, we just use two intermediate layers Conv7 and Conv10 for CIFAR-100 and Conv7 and Conv9 for CIFAR-10. These layers are selected based on their accuracy and the intuition behind the previous experiment. In the Resnet version, the gap between the teacher and the student is larger which requires more intermediate layers. Therefore, we select four intermediate layers instead of the two in the Plain version, and following the same reasoning as the Plain version and the results of each layer, we selected the Conv75,77,79,81 for CIFAR-10 and Conv103,105,107,109 for CIFAR-100.

Table 4.6: Comparison of Plain2 and Resnet8 accuracies trained with ground truth labels, KD, and our method with Resnet110 as the teacher.

Model	**Dataset**	**Baseline**	**KD**	**LPKD**
Plain	CIFAR-100	44.39	49.54	53.3
	CIFAR-10	73.00	74.06	75.49
Resnet	CIFAR-100	57.97	60.98	61.56
	CIFAR-10	85.68	87.47	88.28

Although LPKD requires more training time than the conventional knowledge distillation, it achieves a significant improvement in the accuracy.

4.4.5 Comparison with other KD methods

Regarding the various knowledge distillation based methods in the literature, we followed [17] to compare our method with the previous ones. Following the same settings, we used the numbers in [17, 8] which is shown in Table 4.7. FT [20] proposed to use the intermediate features in both networks. AT [24] uses activation and gradient-based spatial attention maps. FSP [23] generates the flow of solution procedure matrix and the student is trained to make a similar matrix. BSS [8] uses boundary supporting sample to focusing on transfer the decision boundary to the student. MTL [26] proposed mutual learning which trains both the teacher and the student in an interactive method in which each of the networks guides the other one. RCO [11] also trains both networks simultaneously but the direction is just from the teacher to the student. Finally, TAKD [17] trains teacher assistant networks as a bridge to transfer knowledge from the teacher to the student.

For our result, we used the provided code of [17] in Pytorch and trained ResNet26 as the teacher for 320 epochs. For training the student, we first used the last three layers before softmax of the teacher as the intermediate layers, followed by the soft targets of the softmax layer and ground truth labels. Training each set of labels for 80 epochs, the whole training procedure takes 400 epochs in total. Although the number of training epochs is higher from the one mentioned in [17], we should mention that our method does not depend on the training of any external network and therefore the comparison could be considered as fair. We also implemented the RCO method in [11] by using the same teacher as ours and training the student 80 epochs for each step similar to our method. We used 80 and 64 epochs as the gap in RCO and the higher accuracy is reported in the table. Our proposed method achieved better results in comparison with all the previous methods. The most closed results with our are for TAKD and RCO. For the TAKD, the reason is that as it is shown in [5, 17], when the Teacher Assistant is

Table 4.7: Comparison of different KD methods on CIFAR-10. The teacher is ResNet26 for both students.

Student	**NOKD**	**KD**	**FT**	**AT**	**FSP**	**BSS**	**MTL**	**TAKD**	**RCO**	**LPKD**
ResNet8	86.02	86.66	86.73	86.86	87.07	87.32	87.71	88.01	88.61	**88.82**
ResNet14	89.11	89.75	89.72	89.84	89.92	90.34	90.54	91.23	91.28	**91.71**

similar to the teacher, the accuracy of the TA could be higher than the Teacher after knowledge distillation. Therefore, in this experiment, not only the gap between the ResNet14 and ResNet26 is very small, ResNet20 as the TA can get higher accuracy than the teacher which helps TAKD to further improve the result. This shows itself when we use ResNet8 as the student and the gap will be increased which causes smaller improvement in the results of TAKD. In the comparison of our method with RCO, both use a progressive method and the results are close, but the intermediate epochs of the teacher could be in different local optimums and produce different distributions which can cause misguiding for the student during training.

Chapter 5

Concolusion

5.1 Summary of the Thesis

In this work, we focus on a well-known method of transfer learning in convolutional neural networks known as Knowledge Distillation. In this regard, first, we had a review on the literature of the topic by describing different methods, and then we propose the two methods Focalized Teacher and Layer-wise Progressive Knowledge Distillation to improve the results. The principal contribution of this work is in the second method which contains of three part and got better results in comparison with the previous knowledge distillation methods. We also did several experiments on two datasets Cifar10 and Cifar100 to show the effectiveness of each part independently.

5.2 Future Works

Curriculum Learning and Progressive methods showed promising results in several works. In our method, we use a direct approach to use the logits of the middle layers and switching between them. However, the more advanced method can be helpful to improve the final accuracy of the student. For example, instead of switching between the layers for all samples together, one idea is to switch for each sample independently.

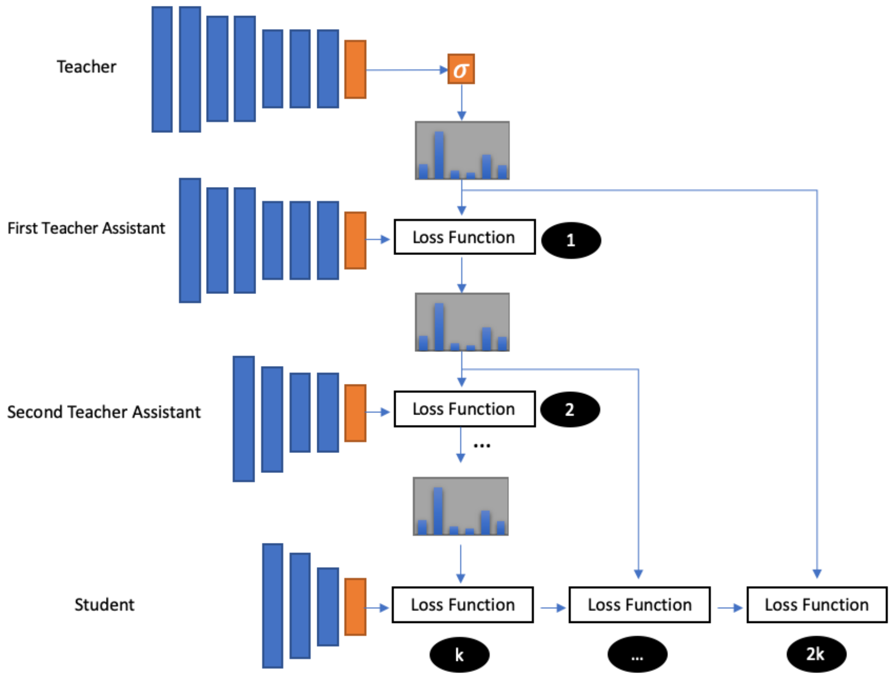


Figure 5.1: Progressive Knowledge Distillation based on the Teacher Assistants could be also a good way for training the student.

The ideas of our method can also be extended to other works, for example, from the aspect of progressive learning, we can also use the teacher assistants instead of the layers to create the progress.

5.2.1 Progressive Teacher Assistant based Knowledge Distillation

Teacher Assistants was proposed to fill the gap between the teacher and the student. However, after completing the procedure by training the student with the soft targets of the smallest teacher assistant, we can use the progressive method on teacher assistants again to reach the teacher's soft target. The idea is shown in Figure 5.1. Although this method needs an exhausting training procedure, it could much better result in comparison with the original TAKD.

Bibliography

[1] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48. ACM, 2009.

[2] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 742–751. Curran Associates, Inc., 2017.

[3] Wenlin Chen, James T. Wilson, Stephen Tyree, Kilian Q. Weinberger, and Yixin Chen. Compressing neural networks with the hashing trick. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, International Conference on Machine Learning (ICML), pages 2285–2294. JMLR.org, 2015.

[4] François Chollet et al. Keras. https://keras.io, 2015.

[5] Tommaso Furlanello, Zachary C Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. *International Conference on Machine Learning (ICML)*, 2018.

[6] Mengya Gao, Yujun Shen, Quanquan Li, Chen Change Loy, and Xiaoou Tang. Feature matters: A stage-by-stage approach for knowledge transfer. *arXiv preprint arXiv:1812.01819*, 2018.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[8] Byeongho Heo, Minsik Lee, Sangdoo Yun, and Jin Young Choi. Improving knowledge distillation with supporting adversarial samples. *arXiv preprint arXiv:1805.05532*, 2018.

[9] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015.

[10] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[11] Xiao Jin, Baoyun Peng, Yichao Wu, Yu Liu, Jiaheng Liu, Ding Liang, Junjie Yan, and Xiaolin Hu. Knowledge distillation via route constrained optimization. *CoRR*, abs/1904.09149, 2019.

[12] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[13] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[14] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.

[15] Tsung-Yi Lin, Priyal Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *IEEE transactions on pattern analysis and machine intelligence (TPAMI)*, 2018.

[16] Microsoft-Research. Neural network intelligence toolkit. 2018.

[17] Seyed-Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant: Bridging the gap between student and teacher. *arXiv preprint arXiv:1902.03393*, 2019.

[18] Antonio Polino, Razvan Pascanu, and Dan Alistarh. Model compression via distillation and quantization. In *International Conference on Learning Representations (ICLR)*, 2018.

[19] Ilija Radosavovic, Piotr Dollár, Ross Girshick, Georgia Gkioxari, and Kaiming He. Data Distillation: Towards Omni-Supervised Learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[20] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *CoRR*, abs/1412.6550, 2014.

[21] Timo Aila Samuli Laine. Temporal ensembling for semi-supervised learning. *International Conference on Learning Representations (ICLR)*, 2017.

[22] Chenglin Yang, Lingxi Xie, Siyuan Qiao, and Alan Yuille. Knowledge distillation in generations: More tolerant teachers educate better students. *ASSOCIATION FOR THE ADVANCEMENT OF ARTIFICIAL INTELLIGENCE (AAAI)*, 2019.

[23] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4133–4141, 2017.

[24] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *International Conference on Learning Representations (ICLR)*, 2017.

[25] Lei Zhang, Peng Wang, Chunhua Shen, Lingqiao Liu, Wei Wei, Yanning Zhang, and Anton van den Hengel. Adaptive importance learning for improving lightweight image super-resolution network. *arXiv preprint arXiv:1806.01576*, 2018.

[26] Ying Zhang, Tao Xiang, Timothy M. Hospedales, and Huchuan Lu. Deep mutual learning. *CoRR*, abs/1706.00384, 2017.

초 록

지식 증류 (Knowledge Distillation, KD)는 교사로부터 학생 모델로 지식을 전달하는 잘 알려진 방법입니다. 본 논문에서는 계층 적 진보적 교사 (Layer-wise Progressive Teacher)를 도입하여 지식 증류를위한 새로운 틀을 제안하고자한다. 이와 관련하여 우리는 교사의 중간 계층에서 확률을 구함으로써 서로 다른 경도 수준에서 부드러운 목표를 만드는 방법을 제안합니다. 우리의 방법은 교사와 학생 사이에 큰 차이가있어 학생이 교사를 모방하는 것을 더 어렵게하는 경우를 위해 특별히 고안되었습니다. 우리는 또한 학생의 온도를 제거하고 교사의 온도를 유지하는 것이 좋습니다. 실험 결과는 기존의 증류법과 비교할 때 우리의 방법이 훨씬 더 우수한 결과를 얻음을 보여줍니다.

주요어: 이미지 분류, 이적 학습, 지식 증류

학번: 2017-28825

ACKNOWLEGEMENT

I would like to express my sincere gratitude to my advisor Prof. Kyoung Mu Lee for the continuous support of my study and research, for his patience, motivation, enthusiasm, and immense knowledge. I could not have imagined having a better advisor and mentor for my Masters' study.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Nam Ik Cho and Prof. Bohyung Han for their consideration, encouragement, and insightful comments.

Last but not least, I would like to thank my family, especially my father and my mother, for all the supports and patient throughout my life.