Neural Network Distillation

Leon Zhang
MIDS
Duke University
lz198@duke.edu

Jeremy Zeng
MIDS
Duke University
xz301@duke.edu

Zihao Lin
ECE
Duke University
zl293@duke.edu

Abstract

In recent years, model distillation has received considerable attention on model compression. In this project, we examined various model distillation mechanism including teacher-student based knowledge distillation, temperature effect on distillation performance, knowledge transfer unseen label, reversed distillation, and self distillation. We conducted several experiments using MLP on MNIST and ResNet CIFAR-10 datasets. The main findings are: using model distillation can improve the performance of smaller model; Low temperatures is more beneficial to the performance; When randomly omit one class of data from dataset, the performance drops rapidly especially on the omitted class of data; Both reversed distillation and self distillation can improve student model's performance, while self distillation can improve more than reversed distillation.

1 Background

Deep neural networks have achieved tremendous success in many fields with the advances of new architecture and improved hardware. As the field progresses, deeper models with increasing parameters are being invented. This introduces new problems such as energy usage and latency restrictions. Many researchers are coming up with different ways to reduce model size while retaining neural networks' superior performance. One popular strategy of model compression is called knowledge distillation.

A typical distillation follows these steps: Feedforward the input to both the large teacher model and the smaller student model, and compute the loss on the softmax output and the ground truth instead of the ground truth alone. The softmax function is also tweaked by adding a hyperparameter called "temperature" which relieves the issue of skewed relativeness of the original softmax [1]. This provides the student with a more accurate probability distribution of the class predictions. Recent research also explores new distillation methods such as self distillation, which we will investigate in this project as well.

In this project, we aim to explore using neural network distillation techniques to compress large size, high performing models for computer vision into smaller, faster models while doing our best to preserve its performance. More specifically, we will train a large teacher model to allow the small student model to learn from the teacher model's soft labels [1]. We will then try reversing the teacher and the student, and finally, we will experiment with self-distillation, which lets the student model learn from its soft label [2]. We speculate that using distillation, a small neural network can achieve better performance than just directly training on the ground truth.

2 Method

We follow the paper [1] to define our knowledge distillation method. We trained a large model and a smaller model separately. Then we used the smaller model as the student model and the large model

as the teacher model to perform knowledge distillation. The general training pipeline is illustrated in Figure 1.

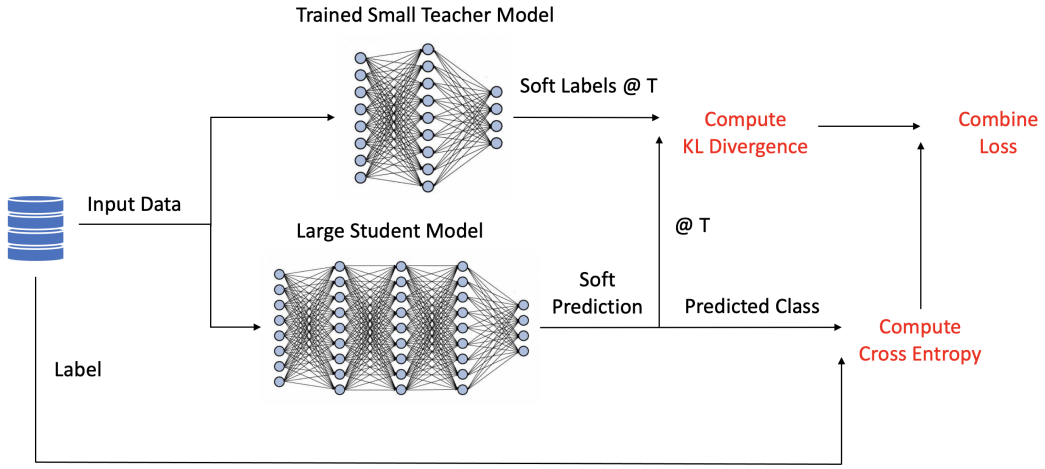

Figure 1: Distillation training pipeline

Instead of using cross-entropy loss usually performed in separate training, knowledge distillation modifies the loss function to be a combination of KL divergence of the soft labels and the cross-entropy loss of the hard label. The Equation below describes the loss function:

$$l = \alpha * \mathrm{KL}(P_{\text{student}}, P_{\text{teacher}}) + (1 - \alpha) * \mathrm{CE}(P_{\text{student}}, Y), \tag{1}$$

where KL stands for KL divergence, CE stands for cross-entropy loss, and α represents the weighting between the two term ranging from 0 to 1. We use q_i to show the probability distribution after softmax and z_i to show the logits feeded into softmax function. The probability distribution is

$$q_i = \frac{\exp\left(z_i/T\right)}{\sum_j \exp\left(z_j/T\right)}, \tag{2}$$

where T is temperature, which is used to soften the probability distribution over classes. Following the paper [1], we set T to be 1 for both student and teacher models. In order to remove the influence of temperature when computing derivative, we multiply the equation by T^2. Therefore, the final loss function becomes

$$l = \alpha * T^2 * \mathrm{KL}(P_{\text{student}}, P_{\text{teacher}}) + (1 - \alpha) * \mathrm{CE}(P_{\text{student}}, Y). \tag{3}$$

3 Experiment results

In the first section of our project, We first trained a large neural net as the teacher model and a small neural net as the student model, and also as a baseline on the MNIST dataset. We then retrained the smaller student model using the knowledge distillation technique to compare its performance. We also tried different values of hyperparameter "temperature" to analyze its influence on the distillation performance.

In the second section, we trained ResNet20 and ResNet50 on the CIFAR10 dataset to explore reverse distillation and self distillation.

3.1 Dataset

For this project, we explored two different datasets to evaluate distillation techniques. In particular, we picked MNIST for knowledge distillation, testing different temperature ranges, and omitting

samples in the dataset. We picked the CIFAR10 dataset to explore more advanced distillation methods including reversed distillation and self distillation

The MNIST dataset contains 70000 handwritten digits from 0 to 9 with 60000 training images and 10000 test images. Half of the training set and half of the test set were merged from the NIST's original training dataset.

The CIFAR-10 dataset consists of 60000 32x32 color images in 10 classes, with 6000 images per class. The 10 classes include airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. The classes are mutually exclusive, meaning that no class can be a subclass of another. For example, automobiles only include sedans, SUVS, etc, but no trucks [3].

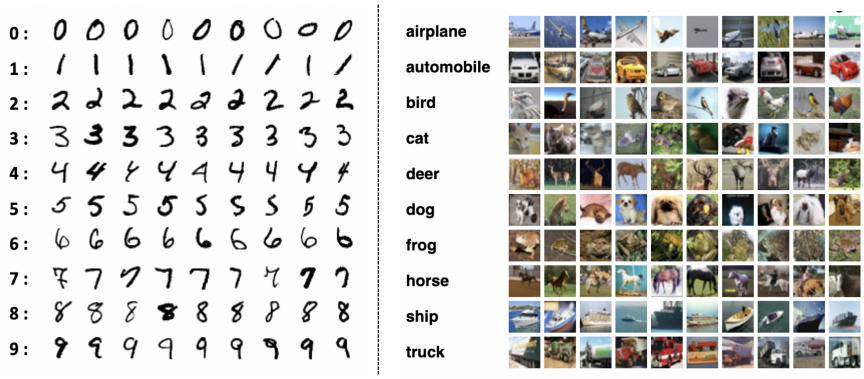

Figure 2: Dataset. The left part shows some examples of MNIST. The right part shows some examples of CIFAR-10.

3.1.1 Knowledge Distillation

We trained a single larger neural net which has two 1200-dimension hidden layers followed by a dropout layer with dropout rate 0.3; and a smaller neural net consist of one 400-dimension hidden layer without dropout layer on MNIST dataset. To provide a sense of model size, the FLOPS of the larger model is 4785600 while the FLOPS of the smaller model is 635200, which is around 7 times smaller. The result is shown in Table 1.

Model	Accuracy
Teacher (L)	0.9887
Student (S)	0.9857
Student (S) + KD	0.9863

Table 1: MNIST: Model Results

We can see that the test accuracy of the teacher (larger, noted as (L) in the table) model is 0.9887, and the test accuracy of the original trained student (smaller, noted as (S) in the table) model is 0.9857.

We then tried knowledge distillation with alpha to be set to 0.7. The test accuracy of the student trained by knowledge distillation is 0.9863, which is a slight improvement from the separately training the small model.

3.1.2 Different Temperature

Temperature in knowledge distillation has the affect of softening the class probability distribution. Higher temperature results in class probability being more equal. We investigated the affect of temperature on distillation performance. We tried a range of temperatures from 0.1, 0.2 to 1; 2, 3 to 10, 20 to 100. In total 30 different temperatures. Figure 3 shows the test accuracy for each temperature.

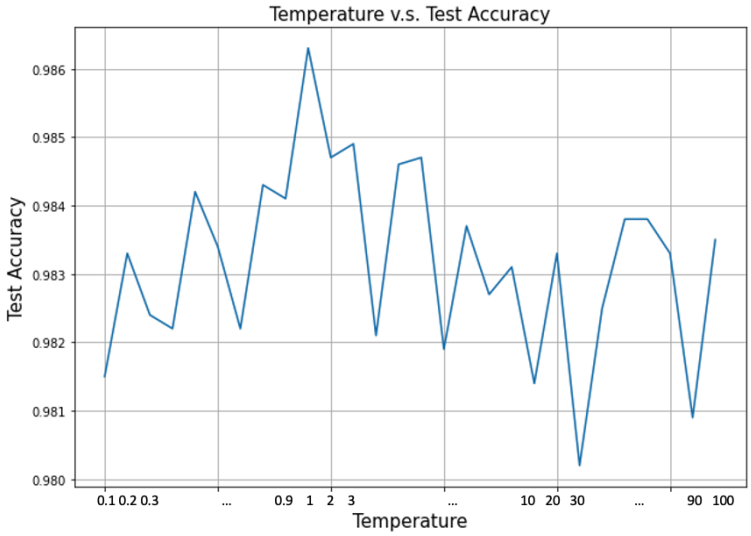

Figure 3: Temperature v.s. Test Accuracy

We observed that with the increase of temperatures from 0.1 to 1, the test accuracy has an increasing trend and reaches the max. However, when we continue increasing the temperature, from 1 to 10 step by 1, and from 10 to 100 step by 100, we can see that the test accuracy decreases.

3.1.3 Omit One Digit

We omitted the digit 3 from our training dataset as a transfer dataset. And we still used the larger single neural net which trained on the original MNIST dataset as the teacher model, and then performed the distillation on the transfer dataset with the smaller neural net. With the full validation dataset, we found the test accuracy of the student trained with the transfer dataset is 0.8868. There are 1132 wrong predictions. While among those wrong predictions, there are 89% cases having label 3. That makes sense because we omit the digit 3 in the training dataset. For all other numbers, the model has similar ability to predict.

3.2 ResNet

Although the CIFAR-10 dataset and MINIST dataset both contain 10 classes, working with CIFAR-10 is considered to be a much harder task since the object in the dataset is more complex and the image is colored.

Due to the increased complexity of the task, we have to use more advanced model architecture to test out distillation techniques. We selected ResNet20 and ResNet50 as our two models. We first trained them separately with the same configuration.

The training step first randomly split the CIFAR-10 dataset into training with 47500 images, validation with 2500 images and test set with 10000 images. We also implemented the data augmentation to improve the model performance by applying random horizontal flipping with 0.5 probability, random cropping with 4 paddings and image normalization as well. We then trained both ResNet20 and ResNet 50 with cross entropy loss for 150 epochs and 128 batch size and initial learning rate 0.01. Besides, we also implemented learning rate decay with 0.1 factor at 75 and 110 epoch, momentum set to 0.9, and L2 regularization set to 1e-4 to help the model optimization and generalization.

In this section, we train different models to see the performance of knowledge distillation. The training result is shown in Table 2.

We can see that the plain-trained ResNet50 model achieved a test accuracy of 0.9079 and the plain-trained ResNet20 model achieved a test accuracy of 0.8967.

Model	Accuracy
ResNet50	0.9079
ResNet20	0.8967
ResNet50 + Reverse KD	0.9128
ResNet50 + Self KD	0.9238

Table 2: CIFAR-10: Model Results

These individually trained models will serve as a baseline and be used to investigate reverse distillation and self-distillation in future sections.

3.2.1 Reverse Teacher/Student Model

The research paper [4] proposed a new take on the knowledge distillation framework. The paper suggests that the teacher does not necessarily have to be a better model. It argues that a large model can also learn from a smaller model even though the smooth label generated from the smaller teacher model is not as good.

We investigated this idea using ResNet20 as the smaller teacher model and ResNet50 as the student model. The general training pipeline is shown in Figure 4.

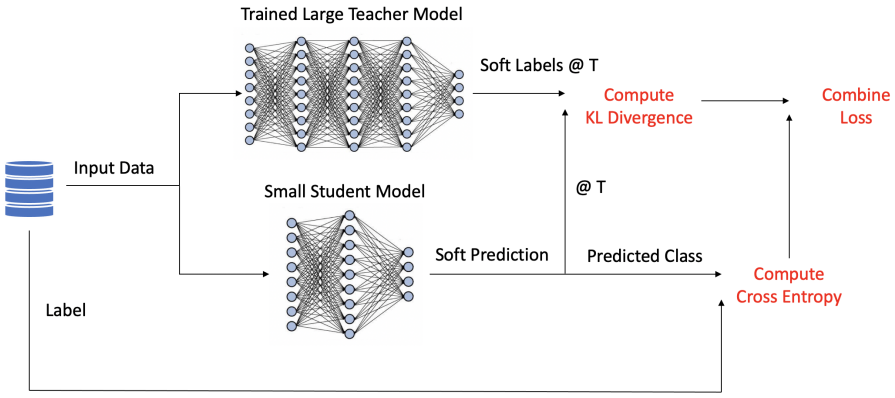

Figure 4: Reverse Distillation

We picked alpha to be 0.1, so the KL divergence loss has weighting 0.1 and the cross entropy has weighting 0.9. We also picked the temperature to be 1, retaining the exact soft label from the teacher model. At last, We chose the same training configuration as the Resnet50 being trained individually. This would help us easier to compare whether reverse distillation helped improve the performance.

We can see from the table 2 that the final test accuracy of the reverse distilled ResNet50 model is reported to be 0.9128, which is a 0.049 performance boost from training RestNet50 separately.

3.2.2 Self Distillation

The paper [4] also proposed self distillation can also result in better model performance. Self distillation is defined as using the trained version of a model to perform distillation on the same model. To test out this method, we used ResNet50. The general training pipeline is illustrated in Figure 5.

We again picked alpha to be 0.1, so the KL divergence loss has weighting 0.1 and the cross entropy has weighting 0.9. We also picked the temperature to be 1, retaining the exact soft label from the teacher model. We also chose the same training configuration as the Resnet50 being trained individually. This would help us easier to compare whether self distillation helped improve the performance.

We can see from the table 2 that the final test accuracy of the self distilled ResNet50 model to be 0.9238, which is a 0.159 performance increase from training ResNet50 separately.

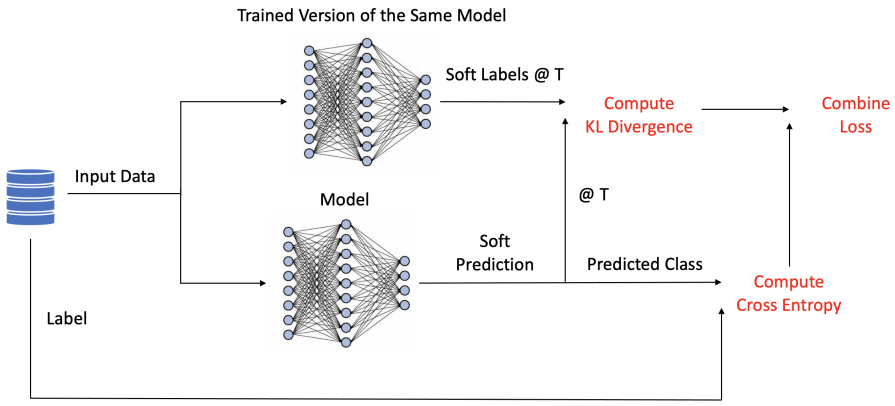

Figure 5: Self Distillation

4 Conclusions

We observed that normal knowledge distillation improved the performance of our smaller student model by increasing 0.0006 accuracy on MNIST dataset. Although the improvement is not significant, we can still notice that knowledge distillation helps improve the smaller student model. One of the possible explanations for the weak improvement is that the student model is robust enough on the MNIST dataset.

We also observed that the test accuracy of the model increases and then decreases while the temperature increases and hits its best accuracy when temperature is set to 1. This makes sense because when T increases, the soft labels are more similar to uniform distribution which means it contains less valuable information, while when temperature becomes too large, the soft label converges to uniform distribution which contains less information resulting in lower test accuracy.

We observed that both reversed and self distillation improved our model's performance given the same training configuration. For reversed distillation, the ResNet50 model improved its test accuracy by 0.049 and for self distillation, ResNet50 improved its test accuracy by 0.159.

This is a surprising finding and it shows a model can still improve itself with distillation and it is relatively insensitive to the quality of the teacher's softlabel. With these observations, we dived deeper into paper and there are two possible explanations.

First possible explanation is that distillation can help models converge to flat minima which features in generalization inherently. Second possible explanation is that distillation prevents models from vanishing gradient problems. We also believe that using soft labels instead of hard labels offers a less extreme way to compute loss, as a result, it provides an additional way to regularize the model.

References

[1] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[2] Li Yuan, Francis EH Tay, Guilin Li, Tao Wang, and Jiashi Feng. Revisit knowledge distillation: a teacher-free framework. 2019.

[3] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[4] Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3713–3722, 2019.