Modification and Experiments of Neural Network Distillation

Minke Yu
Duke University
minke.yu@duke.edu

Ziling Yuan
Duke Uiversity
ziling.yuan@duke.edu

Abstract

From Hinton's[1] proof and practice, using distillation technology, knowledge of integrated models can be compressed into a single model, making it easier to deploy. Based on the MNIST and Cifar10 data sets, this study investigated the validity of the Distillation model, the optimization of the distillation model (Temperature adjustment and loss function selection), and the validity of the variant of the distillation model (Reversed Distillation; Distillation using incomplete transfer dataset; Self-Distillation) was explored and discussed. The experiments can be viewed on our github page: `https://github.com/ymkymkymkymx/ECE661_knowledge_distilation_project_minke-ziling`

1 Introduction

There has been considerable growth in the field of deep learning in recent years, with advances in computer vision, natural language processing and other fields. Training different models on the same data set and then integrating their predictive capabilities can improve the performance of machine learning algorithms. However, the problem with model integration is that it is computationally expensive and difficult to deploy on a large scale.To solve this problem, researchers have come up with techniques to compress or extract knowledge of complex models to make them smaller and simpler.

We find that as for SimpleNN on MNIST and resnet on cifar10, distillation works better than their baseline. The mechanism is easy to understand: have teacher assign "soft labels" as input data of student instead of hard label so that more knowledge learnt by teacher can be conveyed. The experiment of 'Remove 1 digit' is valuable to see the power of distillation. We try to randomly omit one digit from MNIST as transfer set and repeat the distillation process , it's amazing that he prediction of digit which we omit for student training reaches over 0.95 accuracy, the knowledge is totally from the information of other digits' labels learnt by teacher.

Meanwhile, in the optimizing of Temperature T and loss function, we find KL-Divergence works better than Cosine embedding loss and MSE loss, and also find the trend of accuarcy increase and decrease for T tuning. We discuss the mechanism in the paper's experiment part.

2 Related Works

The studies cited, referenced and reproduced in this paper mainly involve the concept of knowledge distillation and its optimization technology, as well as the concept of self-distillation and its implementation logic.

Preprint. Under review.

2.1 Knowledge Distillation

Knowledge Distillation [1] is an effective technique used for compressing large models for deployment.

As in the Figure1from paper[2], the distillation process involves using the teacher model to generate predictions (or soft labels) for a given input. These soft labels are then used to train the student model instead of using the true labels.The student model is then optimized using the soft labels and trained to produce similar predictions to those produced by the teacher model.The student model is then able to make more accurate predictions as it inherits the knowledge of the teacher model.

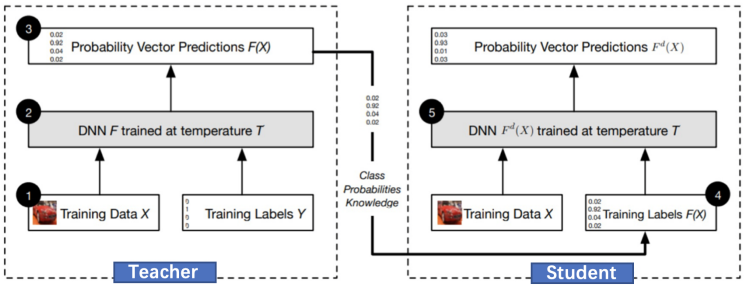

Figure 1: Knowledge distillation structure

2.2 Self-Distillation

The research and mechanism of self-distillation is a process of modifying a model to improve its performance using its own predictions. It is based on the concept of using the same model multiple times and adjusting the parameters to better learn from the data.

In the process of self-distillation, the model structure is adjusted to make the prediction more accurate in a way similar to self-training. This process can also be done in different ways depending on the particular application.

As what is shown in Figure2, according to the framework of Linfeng's paper[3].

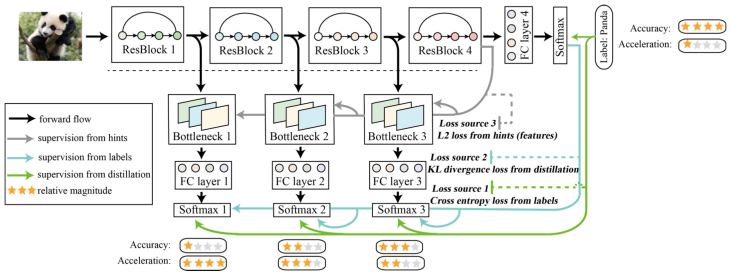

Figure 2: Self-distillation process.

3 Methodology

In this part we will talk in detail about: 1) Distillation on MNIST, how we design our teacher and student models, and the mechanism behind distillation;2)Self-Distillation on cifar10: the way we design the self-distillation model with resnet50 followed by mechanism proposed by LinFeng Zhang; 3)Model modification: Temperature and different loss function: the way we tuned the parameters and selecting the loss functions.

3.1 Distillation on MNIST

The key of Distillation is to use the soft-labels of teacher network instead of hard label of original data as the input of student model. Two simpleNN are designed as the two networks, both of them are composed by 4 layers and has 28*28 dimension as input, and 10 dimension as output(one flatten layer, one linear layer, one Relu layer, and then a linear layer). The teacher network is much more complex with 3X dimension of hidden layer compared with student network.

The two networks' FLOPS, number of parameters and the best accuracy of training alone on MNIST are shown as Table 1. It's obviously to see that Teacher network has much more FLOPS and Parameters number, and has greater best accuracy.

Table 1: Baselines of Teacher and Student Network on MNIST

Network	FLOPS Number	Parameters Number	Best Accuracy
Teacher	33,716,800	33,716,800	0.9928
Student	11,206,400	11,206,400	0.9907

3.1.1 Distillation and Self-Distillation on cifar10

First, as the experiment, we implement resnet10 and resnet10 on cifar10 dataset and doing the distillation work.As for student network, resnet20 is used with FLOPs 28,906,850 and number of parameter 3,490,570; as for the teacher network, resnet20 is used with FLOPs 2.63 G and number of parameter 818,282,as shown in Table 3.

Table 2: Teacher and Student Network on cifar10

Network	FLOPS Number	Parameters Number
Teacher(resnet20)	2.63 G	818,282
Student(resnet10)	28,906,850	3,490,570

Then, we implement self-distillation on cifar10 by resnet50.Follow the framework of Linfeng's paper, we do the following work. ResNet50 was divided into 4 sections according to ResBlocks. Shallow classifiers were set respectively, combined with bottleneck layer (used to reduce the impact of shallow classifiers before) and full connection layer (used only in training and removed during predict).

In addition, the loss function is composed of three parts: 1) the deep and shallow classifier softmax layer predicts the cross entropy loss of tags; 2) KL divergence computed by softmax outputs between students and teachers; 3) L2 loss between feature maps of deep shallow classifier. The formula of Self-Distillation is written as below,which is also from paper[3]

$$loss = \sum_{i}^{C} \left((1-\alpha) \cdot CrossEntropy\left(q^i, y\right) + \alpha \cdot KL\left(q^i, q^C\right) + \lambda \cdot \|F_i - F_C\|_2^2 \right) \quad (1)$$

3.1.2 Model modification: Temperature and different loss function

Temperature(T) is an really useful adjustment parameter of the distillation model used by Hinton. The larger the value of T, the softer target of teacher network output will become smoother (soft).The significance of the temperature coefficient is that by dividing the T term when calculating the softmax loss function, it magnifies the loss value corresponding to the probability value of other classes, and magnifies the contribution of the error result with small probability to the loss function. In this way, different kinds of output results are equally considered, which promotes the training effect of student network.

4 Experiments

In this part, we give the result of all the experiments we implement and analyze the mechanism behind the result.

4.1 Distillation on MNIST

In this experiment, we test the performance of baseline teacher network and student network, and also the distilled student model.We tried to train the small network on MNIST using soft tags generated by the large model and compared its performance to that of the same small model trained using ground truth. From Table 3, we find that After learning the output of teacher network, distilled student network's accuracy increased from 0.9907 to 0.9921, much closer to the accuracy of teacher network.

Table 3: Comparison of Distilled Model and Baseline on MNIST

Model	Best Accuracy on MNIST
Distilled Student Model	0.9921
Baseline Student Model	0.9907

4.2 Model Modification with Temperature and Loss functions

As we discussed in Methodology part, temperature(T) is a really useful adjustment parameter. We adjust the T parameter of the distillation model with KL divergence, cosine embedding and MSE as the loss function. In the experiment, We try different temperatures for thses three types of loss function as shown in Figure3. We find that the Validation Accuracy first increase, arrive at the top around then decrease. It make sense because we need to find a T that get the most suitable softness of label, so that student network can get the most from transfer dataset.

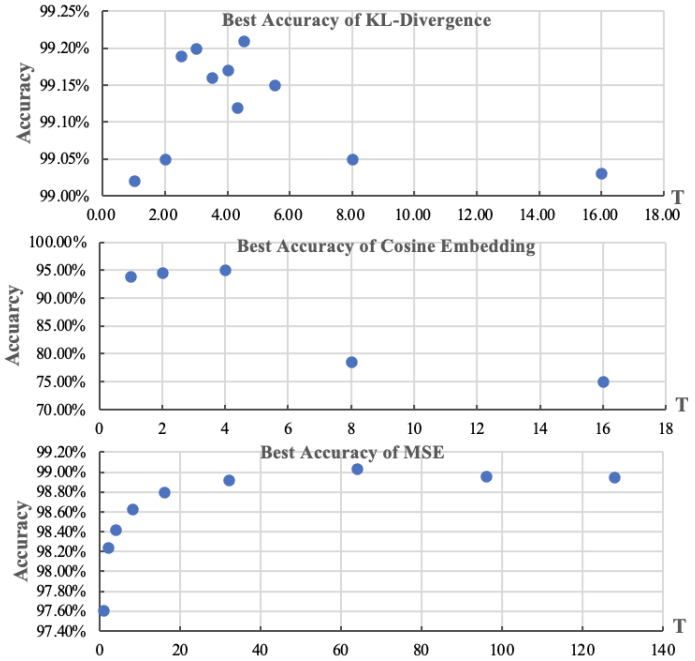

Figure 3: MNIST result

We also do the loss functions selection and T parameter tuning for distilled resnet model on cifar10. The results are given in the Figure 4.The KL-Divergence loss still do the best job when T is around

Table 4: Distilled Models with different loss function on MNIST

Loss	Best Accuracy
KL Divergence loss	0.9921
Cosine Embedding loss	0.9499
MSE loss	0.9903

4 to 5 with Best accuracy 0.8906. As for MSE loss, its best T is around 2 to 3 with best accuracy 0.8856. It is obviously to see that cosine embedding loss does a bad job.

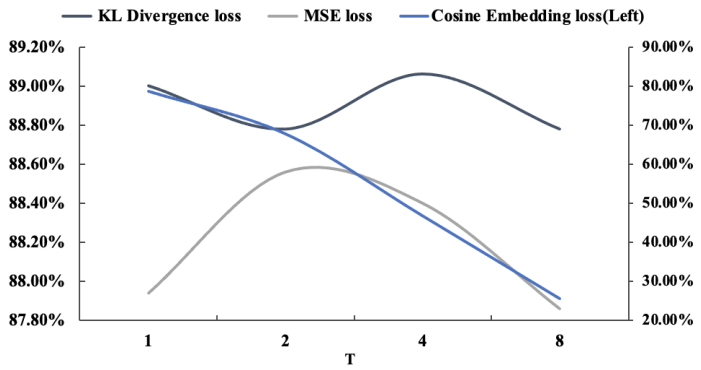

Figure 4: CIFAR10 result

The possible reason why KL-Divergence always do the best are given as follow: KL-Divergence loss is able to capture the differences between softmax distributions more accurately than other loss functions, thus making it more suitable for distillation tasks such as those seen in simpleNN and resnet. Additionally, KL-Divergence loss is less sensitive to outliers than mse loss, making it more robust to noisy data.

Cosine embedding loss is not a suitable loss function for distillation probably because the cosine information of the vector is not useful when temperature is high. when the temperature is high, the output tends to be softener and the direction information for cosine loss is not robust and hard to learn.

4.3 Distilled Model with one digit training data removed

This experiment reveals the power of Distilled model.Omit one digit from MNIST dataset as transfer set and repeat the distillation process, the student network can still reach the accuracy around 0.99. The most meaningful aspect in this experiment is that even with no train dataset with the omited digit label, the student can still predict it with over 0.95 accuracy. It means that the student is learnig this knowledge from the soft label of other digit data. The experiment results are what are shown in this Figure6. The reason the case where not complete transfer train set works better than complete may be the digit 0 and 1 have more confusing data which may do harm to the prediction of the model.

4.4 Self-Distillation on cifar10

Self-Distilled Resnet50 has higher validation accuracy than baseline on Cifar10. To some extent proved the efficiency of self-Distillation"s logic, which is learn from itself for higher performance.Thec comparison of Best validation accuracy of ResNet50 Self-Distilled on Cifar10 and ResNet50 Baseline on Cifar10 are given in 3. We have also done the ablation test where we remove the self-distillation from training and keep all other hyper parameters as the same as the 0.9052 model. The accuracy of normal training is 0.886 which is still a lot worse than our self-distillation result.

Table 5: Self-Distillation for ResNet50

Network	Best Accuracy
ResNet50 Self-Distilled on Cifar10	0.9052
ResNet50 Baseline on Cifar10	0.875

Self-distillation works better than the network itself in the situation where Resnet50 is used on dataset CIFAR10. The reason may be self-distillation allows the model to regularize its early layer outputs and made earlier layers producing more meaningful features for the last layer to learn.

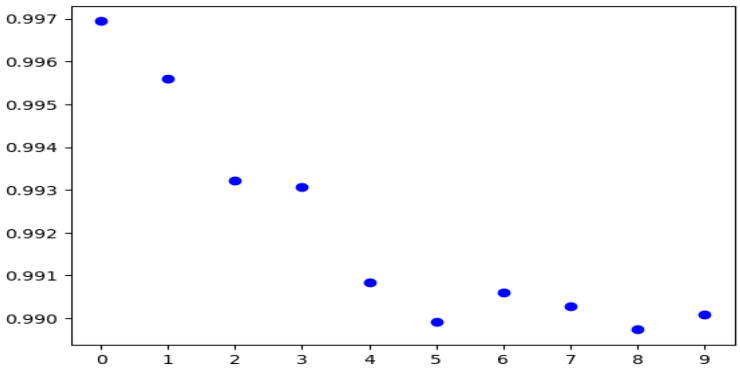

Figure 5: Accuracy for each label before removing the training data.

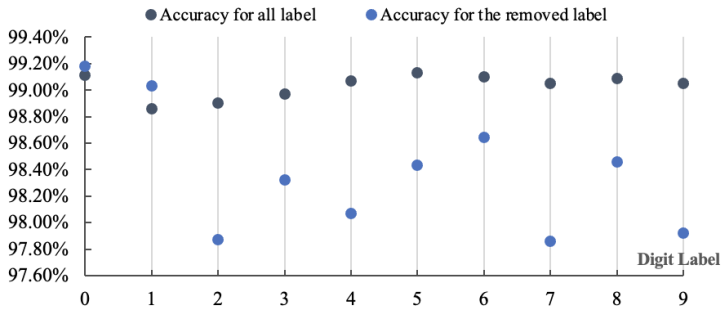

Figure 6: Accuracy with 1 label removed while training

4.5 Reverse the teacher and student network of distilled model

We also try to reverse the teacher and the student model, which means now the student model is larger than teacher model. We run the experiment on CIFAR10,use Resnet11 with accuracy 0.8968 as the teacher and Resnet20 as the student. The baseline for Resnet20 is 0.9188 and with the reverse distillation, it indeed reached a better result which is 0.925. The experiment is shown as Table 6.

Table 6: Reversed Distilled Model on Cifar10

Network	Best Accuracy
Teacher(Previous Student)	0.8968
Student(Previous Teacher) Distilled	0.925
Student(Previous Teacher) Baseline	0.9188

5 Conclusion

The significance of distillation is to transfer the knowledge of the complex teacher model to the lightweight student model by training with soft label, thus greatly improving the prediction accuracy of the student model. In terms of 1) The optimization of the distillation model, taking KL-Divergence as the loss function and selecting T = 4.5, our distillation model had the best performance in MNIST, reaching 0.9921 validation Accuracy. 2) Variation of distillation ModelDistilled Model with one digit training data removed, the prediction of digit which we omit for student training reaches over 0.95 accuracy, the knowledge is totally from the information of other digits' labels learnt by teacher, which proved the meaning of Distillation, Reversed Distillation makes sense, and self-Distillation of resnet50 has a better than baseline performance on cifar10.

References

[1] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. "Distilling the knowledge in a neural network". In: *arXiv preprint arXiv:1503.02531* 2.7 (2015).

[2] Nicolas Papernot et al. "Distillation as a defense to adversarial perturbations against deep neural networks". In: *2016 IEEE symposium on security and privacy (SP)*. IEEE. 2016, pp. 582–597.

[3] Linfeng Zhang et al. "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*. 2019, pp. 3713–3722.