Self-Distillation in Convolutional Neural Networks: A Comparison with Knowledge Distillation on CIFAR-10

1st Veljko Petkovic
Faculty of Sciences and Mathematics
University of Nis
Nis, Serbia
veljko.petko0022@gmail.com

Abstract*—Knowledge distillation (KD) is a popular technique for model compression where a smaller student network is trained to mimic a larger teacher network. Recently, self-distillation methods have been proposed in which a network uses its own deeper layers as a teacher to guide its shallower layers. In this work, I experimentally compare traditional KD (using a pretrained ResNet50 teacher and a ResNet18 student) versus self-distillation (in the style of Zhang et al., ICCV 2019 "Be Your Own Teacher") on the CIFAR-10 image classification task. All models were trained on Google Colab (NVIDIA L4 GPU). The ResNet50 teacher achieved 90.1% test accuracy in 1h 56m 17s. Distilling knowledge from this teacher into a ResNet18 student achieved 94.2% accuracy in 2h 13m 6s. In contrast, a ResNet18 trained with self-distillation (no separate teacher) reached 94.4% in 2h 13m 52s. Table 1 summarizes the accuracy and training time. We analyze these results: self-distillation helps the network find stable, flexible solutions and learn clearer differences between objects. This makes the model much better at handling new data it hasn't seen before. This study highlights the practical advantages of self-distillation over classical teacher-student KD under these settings.

I. Introduction

Deploying high-performance neural networks on resource-limited devices often requires model compression or acceleration techniques. Knowledge distillation (KD) is one such technique wherein a large teacher model's outputs are used to train a smaller student model. In KD, the teacher is first trained (often achieving high accuracy), and then the student is trained to match the teacher's output distributions (the "soft targets") in addition to the true labels. The soft targets carry "dark knowledge" about class similarities (e.g. the teacher may assign moderate probability to classes that are visually similar). Matching these softened outputs via a loss (commonly KL-divergence) helps the student generalize better than if it were trained on hard labels alone. Seminal work by Hinton et al. (2015) formalized this approach and showed that a student trained in this way often achieves higher accuracy than the same model trained conventionally on labels alone.

A. Self-Distillation Without an External Teacher

While KD uses a separate pre-trained teacher, self-distillation has emerged as a variant where a network essentially teaches itself. In the "Be Your Own Teacher" framework from Zhang, a single network is augmented with intermediate classifiers at various depths. During training, the deepest (final) classifier produces soft outputs that serve as targets for the shallower classifiers. In other words, knowledge is distilled within the network: the deeper layers (as "teacher") supervise the earlier layers (as "students"). After training, the some classifiers are dropped, and only the original network remains for inference. This method requires no external teacher model but has been empirically shown to improve accuracy by over 2% on average.

B. Experimental Comparison of Knowledge Distillation and Self-Distillation

In this paper, we compare traditional KD and self-distillation on CIFAR-10 using ResNet architectures. We train a ResNet50 teacher (90.1% accuracy) and then perform KD to train a ResNet18 student. We also implement self-distillation on ResNet18 following Zhang's method. We report the training time and final test accuracy of each method, and we analyze why self-distillation yields a slight improvement (94.4% vs 94.2%) despite similar compute cost. Our contributions include (1) empirical results comparing KD and self-distillation under equal training conditions, (2) a summary table of time vs accuracy, and (3) an analysis grounded in theory as described by Zhang.

II. Methodology

We performed all experiments on the CIFAR-10 dataset using Google Colab with an NVIDIA L4 GPU. CIFAR-10 consists of 60,000 32×32 color images in 10 classes, with a standard 50,000/10,000 train/test split. We used standard data augmentation (random crop and horizontal flip) and normalized inputs.

Identify applicable funding agency here. If none, delete this.

A. ResNet50 Teacher

We trained a ResNet50 model (architecture from He et al., 2016) on CIFAR-10 with cross-entropy loss (no distillation). Training lasted 200 epochs with an initial learning rate of 0.1, decaying by 10× at epochs 100 and 150. The teacher achieved a test accuracy of 90.1% in 1h 56m 17s. This ResNet50 serves as the fixed teacher for subsequent distillation.

$$p_i = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)} \tag{1}$$

B. ResNet18 Student (Traditional KD)

We used a ResNet18 as the student model. During distillation, the student is trained with a combined loss: the usual cross-entropy loss on the true labels (hard targets) plus a distillation loss between the student's and teacher's output distributions. Specifically, we applied a softened softmax (with temperature T) to the teacher's logits to produce a high-entropy "soft target" distribution. The student's output (at the same temperature) is trained to match this distribution via KL-divergence. In practice we followed standard KD settings: we set $T = 4$ and used a weighting factor so that the KL loss and the hard-label cross-entropy each contribute appropriately. The student was trained for the same number of epochs (200) with identical learning rate schedule. This distillation process took 2h 13m 6s and resulted in 94.2% test accuracy.

$$\mathcal{L}_{KD} = KL(p^{(T)}_{\text{teacher}} \parallel p^{(T)}_{\text{student}}) = \sum_i p_i^{(T,\text{teacher})} \log \frac{p_i^{(T,\text{teacher})}}{p_i^{(T,\text{student})}} \tag{2}$$

$$\mathcal{L}_{\text{total}} = \alpha \mathcal{L}_{CE} + (1-\alpha) T^2 \mathcal{L}_{KD} \tag{3}$$

C. ResNet18 with Self-Distillation

We implemented the self-distillation framework from Zhang on the same ResNet18 architecture. Following "Be Your Own Teacher", we divided ResNet18 into 4 segments. We attached auxiliary classifiers (fully-connected layers) at the end of each segment, so there are four classifiers in total. During training, the deepest (4th) classifier's logits provide the pseudo-label distribution. Each shallower classifier is trained to match the deepest classifier's soft outputs (using a KL-divergence loss) in addition to its own cross-entropy loss on the true labels. This effectively distills knowledge from layer 4 into layers 1–3. After training, only the original ResNet18 (up to layer 4) is retained for evaluation; the auxiliary classifiers are discarded. This self-distillation training took 2h 13m 52s and achieved 94.4% accuracy.

$$p^{(T)}_{\text{teacher}} = \text{softmax}(z_{\text{final}}/T) \tag{4}$$

$$p^{(T)}_k = \text{softmax}(z_k/T) \tag{5}$$

$$\mathcal{L}^{(k)}_{KD} = KL(p^{(T)}_{\text{teacher}} \parallel p^{(T)}_k) \tag{6}$$

$$\mathcal{L}_{\text{total}} = \sum_{k=1}^{K} \left[\alpha \mathcal{L}^{(k)}_{CE} + (1-\alpha) T^2 \mathcal{L}^{(k)}_{KD} \right] + \mathcal{L}^{(\text{final})}_{CE} \tag{7}$$

D. Performance

We compare top-1 accuracy and total training time of each method (ResNet50 teacher, KD student, self-distilled ResNet18) under the above setup. Times are wall-clock on Google Colab (L4 GPU).

TABLE I
PERFORMANCE COMPARISON ON CIFAR-10

Method	Top-1 Accuracy (%)	Training Time
ResNet50 (Teacher)	90.1	1h 56m 17s
ResNet18 (Student, KD)	94.2	2h 13m 06s
ResNet18 (Self-Distillation)	94.4	2h 13m 52s

E. Why student got better results than teacher in KD

Even though the teacher network (ResNet50) has a larger capacity, the student (ResNet18) can achieve higher accuracy after knowledge distillation due to several reasons:

1) **Soft targets carry richer information:** The teacher's softmax outputs provide probabilities over all classes, not just the correct label. These soft targets encode inter-class similarities that guide the student to learn more structured and generalizable features, helping it avoid overfitting to hard labels.
2) **Regularization effect:** Distillation acts as a form of regularization, pushing the student toward flatter minima in the loss landscape. Flatter minima correspond to solutions that generalize better to unseen data, which can sometimes result in the student outperforming the teacher.
3) **Shallower networks can generalize better with guidance:** The student's smaller architecture forces it to focus on the most essential patterns, avoiding redundancy in deeper layers. With the teacher's guidance, it captures the critical features efficiently, sometimes surpassing the teacher that may overfit subtle variations.
4) **Self-distillation reinforces knowledge internally:** In self-distillation, deeper layers act as the teacher for shallower layers. This repeated internal supervision improves feature discrimination and representation learning at all depths, which can slightly boost accuracy beyond the original teacher.
5) **Noise smoothing and feature alignment:** The student's training process under distillation encourages learning smoother decision boundaries and aligning feature representations with the teacher. This reduces variance and improves robustness, which often translates to higher test accuracy.

III. Results

The results are summarized in Table 1. As expected, the ResNet18 student trained with distillation substantially outperformed the ResNet50 teacher (94.2% vs 90.1%), showing that knowledge transfer helped the smaller model generalize. Self-distillation yielded a further slight improvement (94.4% accuracy) on ResNet18 over the traditional KD student. The training times for KD and self-distillation were comparable (2h13m each), only modestly higher than the teacher training time of 1h56m. These times include the overhead of computing the distillation losses (for student and self-distillation) and, in the case of KD, the extra forward pass through the teacher network for every batch. The marginal 46-second difference between KD and self-distillation training time suggests the latter incurs negligible extra cost beyond KD.

The key observation is that self-distillation slightly outperforms traditional distillation on this task. We note that the student models (with or without self-distillation) both surpass the teacher, which is consistent with prior findings that KD-trained students can even exceed their teachers' accuracy. Table 1 encapsulates the trade-offs: student training (with any distillation) took only 15–17% more time than training the teacher, but achieved 4 percentage points higher accuracy. The extra gain from self-distillation (0.2 points) is smaller but non-negligible given identical architecture and compute.

IV. Conclusion

We have presented a systematic comparison of traditional teacher–student knowledge distillation and self-distillation on CIFAR-10 using ResNet models. Our ResNet50 teacher achieved 90.1% accuracy; a ResNet18 student distilled from this teacher reached 94.2%, and a ResNet18 trained with self-distillation attained 94.4%. The self-distilled model obtained the highest accuracy with essentially the same training time as the KD student. This confirms that self-distillation can slightly outperform standard KD under these conditions.

The improvement of self-distillation is supported by theoretical arguments: it biases the model toward flatter minima and sharper feature discrimination, and it eases training via intermediate supervision. These factors contribute to better generalization and effective convergence, explaining the observed accuracy gain. Importantly, self-distillation achieves this without requiring a separate large teacher network.

References

[1] Be Your Own Teacher: Improve the Performance of Convolutional Neural Networks via Self Distillation
https://openaccess.thecvf.com/content_ICCV_2019/papers/Zhang_Be_Your_Own_Teacher_Improve_the_Performance_of_Convolutional_Neural_ICCV_2019_paper.pdf

[2] What is Knowledge distillation? — IBM
https://www.ibm.com/think/topics/knowledge-distillation

[3] Hinton, G., Vinyals, O., Dean, J. Distilling the Knowledge in a Neural Network.
https://ar5iv.labs.arxiv.org/html/1503.02531