Offline Knowledge Distillation using Teacher and Peer Learning

Abstract

It is challenging to deploy large neural networks on systems with limited resources. Knowledge distillation was proposed as an approach to teach a small student model from a large teacher model. Different mechanisms have been proposed to perform knowledge distillation and many approaches are built on the analogy of how knowledge is transferred in a classroom setting. We extend the analogy of Teacher-Student learning further and propose an architecture of Teacher-Student-Peer learning, where knowledge is transferred from a teacher to multiple students as well as among students as collaborative knowledge transfer, similar to what happens in a classroom setting.

1. Introduction

In real-world use cases, Deep Neural Networks have gained significant success with respectable performance because of over parameterization and large scale architecture. However, deploying these models on small scale mobile and remote devices is a difficult task due to limited computing power and memory of these devices. The main idea of Knowledge distillation is that the student should mimic the teacher model to achieve a comparable or even better performance than the teacher. The smaller network tries to replicate the outputs of the bigger network. Our proposed idea plans to experiment on the Teacher-Student-Peer architecture which will be an extension of the above Teacher-Student architecture. Our main objective is to evaluate our proposed Knowledge Distillation method against the vanilla variant by using the CIFAR-10 dataset.

Following plan was followed to approach the problem:

1. Refer resources for existing implementations of the original vanilla Teacher-Student model.

2. Implement Knowledge Distillation between a Teacher and multiple students (Student Ensemble). Try out different architectures for student models in the ensemble.

3. Define a loss function for each student, which will be a weighted combination of its cross-entropy loss and distillation loss.

4. Experiment with different ways to construct the distillation loss for students using teacher's and peer's knowledge.

2. Related Work

There has been a myriad of strategies proposed to effectively perform knowledge distillation. Different architectures like Teacher-Student, Multiple Teacher, Teaching Assistant; Different knowledge like Response, Features, Relations; Different Schemes like Offline (teacher teaches student), Online (teacher learns with student) have been explored. The paper by Gou *et al*. [2] provides an comprehensive survey of recent Knowledge Distillation techniques. Work has been done on Teacher-Student learning by Hinton *et al*. [3] and its multiple variations as shown by Gou *et al*. [2]. Peer learning has been explored too, but only in an online setting (in scenarios where there doesn't exist a large-capacity, high-performance teacher) by Wu *et al*. [9]. Our proposal tries to combine these two settings and create an architecture that fosters a student to distill knowledge from teacher and peers simultaneously.

Bucilua *et al*. presented a way of compressing large ensembles into smaller models without significant drop in performance to facilitate deployment of these models on mobile devices [1]. The knowledge transfer between a fully-supervised teacher model and a student model using the unlabeled data is explored for semi supervised learning by Urner *et al*. which showed that the knowledge transfer from teacher to student model using unlabeled data is PAC learnable [6]. Knowledge distillation has also been assessed for label smoothing, to compute the accuracy of the teacher and for getting a prior for the optimal output layer geometry by Tang *et al*. [4]. According to Mirzadeh *et al*. the student network performance degrades when the gap between student and teacher is more. They suggest that a teacher can effectively transfer its knowledge to students up to a certain size. To overcome this they introduced multi-step knowledge distillation which employed an intermediate-sized network called teacher assistant to bridge the gap between the student and the teacher [7]. Inspired by knowledge distillation technique, the idea has been further applied in com-

pressing the training data which transfers the knowledge from a large dataset into a small dataset to reduce the training load of deep models by Wang *et al*. [8]. There is a recent survey on knowledge distillation, which presents the comprehensive progress from different perspectives of teacher-student learning for vision and its challenges by Yoon *et al*. [5].

3. Approach

We are dividing the implementation of the experiment into three parts:

3.1. Training a High-Performance Teacher

The first step is to train a teacher model. A teacher model can be any high performance model. The teacher will be the primary guide for the student models. Its role is similar to a teacher in a vanilla Teacher-Student Knowledge Distillation approach.

3.2. Training the Teacher-Student Pairs

As our baseline we trained three pairs of teacher and student. The models were trained using the vanilla Teacher-Student knowledge distillation approach (code has been adapted from the example of knowledge distillation in keras[1]). The approach for the training and evaluating our baselines are as follows:

1. Select a student model to train.
2. We define a student loss function as the difference between student predictions and ground-truth using the Cross-Entropy loss.
3. We define a distillation loss function, along with a temperature of 10, on the difference between the soft student predictions and the soft teacher predictions using the KL Divergence loss.
4. An α factor to weight the student and distillation loss. We select the α to be 0.1 for our baselines.
5. We optimized the loss using Adam optimizer for 5 epochs.
6. To compare, we trained each student model from scratch using the Cross-Entropy loss and optimized using Adam optimizer for 5 epochs.

3.3. Training the Teacher-Student-Peer Ensemble

For training our proposed model of Teacher-Student-Peer ensemble, we are following the below approach,

1. Select the trained high-performing teacher model.
2. Select the student models.
3. Each input image is predicted independently by the teacher and each student model. For each student, we define the loss using the below template 1,

$$L = \alpha * S + \beta * TS + \gamma * (P1 + P2) \qquad (1)$$

where,

   (a) S is the Student loss. It is the Cross-Entropy Loss between the student's prediction and the ground-truth.

   (b) TS is the Teacher-Student loss. It is the KL Divergence loss between the soft prediction of the student and the teacher.

   (c) $P1$ and $P2$ are the Peer loss. It is the KL divergence loss between the soft prediction of the student and its peers, in this case the other two students.

   (d) α, β, γ are the weights assigned to the respective part of the losses to define the final loss.

4. We train each student using its loss function. So each student contributes in defining the loss of its peers and the students learn from their loss.

We experimented with three different ways to build this loss,

Fixed Loss We define a loss where the α, β, γ are fixed with 0.1, 0.8 and 0.1 respectively. The idea was that the student should mimic the teacher as closely as possible, but also learn from mimicking its peers.

Hyperparameter tune the loss parameters We define a loss where the α, β, γ are tuned using hyperparameter tuning. The idea was to tune what the best configuration should be to teach the students. For tuning we tried with two objectives:

1. Which gives the highest accuracy for the weakest student (model which has the lowest accuracy).
2. Which gives the highest accuracy for the strongest student (model which has the highest accuracy).

Alternate Learning Here the students will learn alternately. This setting can be thought of as a set of teaching-assistants or seniors teaching the student along with the teacher. The idea is to evaluate how the student learns when it is mimicking the teacher and its peers who have already learned from the teacher. Concretely, if we label the students as S1, S2, and S3, the setting will be:

[1] https://github.com/keras-team/keras-io/blob/master/examples/vision/knowledge_distillation.py

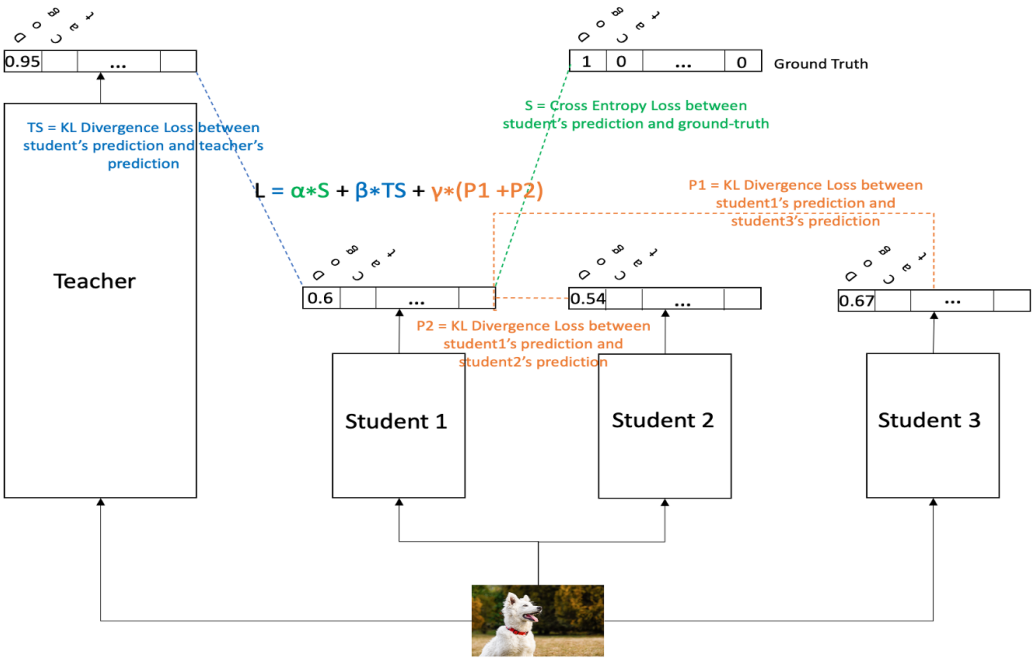

Figure 1. The figure defines the loss function in our proposed Teacher-Student-Peer architecture and the ingredients of the loss for a student. Similarly, losses for other students are defined.

1. Teacher teaches S1, S2 using the Teacher-Student-Peer learning and the fixed or hyperparameter-tuned loss and S3 will be out of the picture.
2. Then, the Teacher, S1, and S2 teaches S3 where the loss of S3 will be dependent on comparison with the soft labels from Teacher, S1 and S2 using a similar loss template as defined above.
3. Repeat this learning for S1 and S2.

4. Experiment and Results

4.1. Dataset

We are using the CIFAR-10 dataset for all our experiments. The dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. The dataset is small, yet effective enough for our experiments. Our main objective is to evaluate the proposed Knowledge Distillation method against the vanilla variant. With this objective, a small dataset which is effective for experimentation is preferable.

4.2. Evaluation Metric

We evaluated our experiments against the following metrics:

1. Compute the accuracy of the student model without distillation by training it from scratch.
2. Compute the accuracy of the student model with distillation by using vanilla Teacher-Student approach.
3. Compute the accuracy of the student model with distillation by using our proposed Teacher-Student-Peer model approach, with and without alternate learning.
4. Compare the accuracy obtained from the above three settings and different loss functions of our proposed approach.

4.3. Training a High-Performance Teacher

We selected ResNet-50 model as our teacher. The model is picked from Keras Applications. The model was pre-trained on the ImageNet dataset. We further fine-tuned the model on the CIFAR-10 dataset by upscaling the CIFAR-10 data by a factor of 7 to match the size of the input accepted by ResNet-50. We used SGD optimizer with Categorical Cross-Entropy loss. It was trained for 3 epochs. With transfer learning, we achieved a performance of 94.75% accuracy on the CIFAR-10 test set for the ResNet-50 model. We believe this is a good accuracy for a high-performance teacher.

4.4. Training students from scratch

We selected three student models: AlexNet, LeNet, MobileNet. We selected these ensemble as this represent diversity in performance of the students. For comparison, we

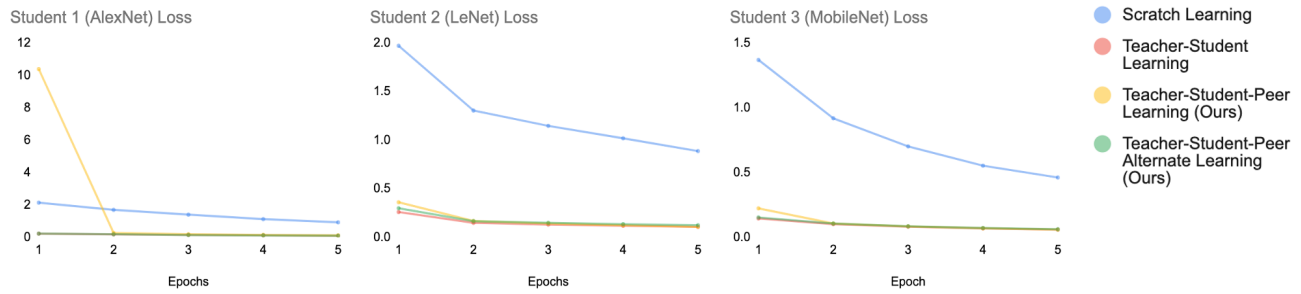

Figure 2. The figure shows the loss over epochs for different approaches: training from scratch, training using the Teacher-Student model, training using the Teacher-Student-Peer model, and training using Alternate learning in the Teacher-Student-Peer model.

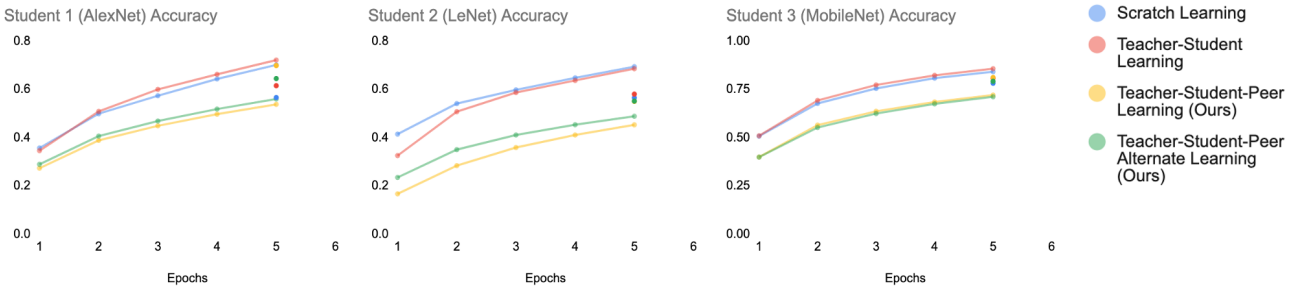

Figure 3. The figure shows the training accuracy over epochs for different approaches: training from scratch, training using the Teacher-Student model, training using the Teacher-Student-Peer model, and training using Alternate learning in the Teacher-Student-Peer model. The test accuracy of the respective setting is shown using the dot marker at epoch 5.

trained each student model from scratch. The models were trained for 5 epochs using Adam optimizer. We have made modifications to the student models to accommodate the input shape of CIFAR-10 data, 32x32x3. It is important to note that the models were not trained using transfer learning, i.e. the weights were randomly initialized before training. MobileNet was the highest performing student with 78.16% accuracy, followed by AlexNet with 60.54% accuracy, followed by LeNet with 56.5% accuracy.

4.5. Training the Teacher-Student Pairs

We trained three Teacher-Student pairs using the vanilla Teacher-Student model approach described in Section 3.2. These are our baselines. We have made modifications to the student models to accommodate the input shape of CIFAR-10 data, 32x32x3. The models were trained for 5 epochs using Adam optimizer.

1. ResNet-50 - AlexNet
2. ResNet-50 - LeNet
3. ResNet-50 - MobileNet

The performance of the student models trained using this vanilla approach can be seen in Table 2. For each of the student models, we see an improvement in accuracy. For AlexNet, we saw an increase of about 0.98% accuracy, for LeNet we saw an improvement of about 1.52% accuracy, and for MobileNet we saw an increase of about 2.59% accuracy from training from scratch. These improvements are absolute increase in accuracy. In Figure 3, we can see that training accuracy gradually increases to match or exceed the performance of training from scratch. However, in test set the performance of Teacher-Student trained students are consistently higher than training from scratch. Thus, it can be seen that this setting induces a regularization effect. Similar effect was observed in the original paper bu Hinton *et al*. [3].

4.6. Training the Teacher-Student-Peer Ensemble

As described in Section 3.3, we experimented with different ways to build the loss function.

4.6.1 Teacher-Student-Peer learning

Here we closely followed our described approach in Section 3.3 and experimented with different values of α, β, and γ. For each hyperparameter, we trained using Adam optimizer for 5 epochs. We splitted the training set to training and validation set using a 80-20 random split. We then selected the model which gave the highest performance in the val-

α	β	γ	Validation Accuracy (%)		
			Student 1 (AlexNet)	Student 2 (LeNet)	Student 3 (MobileNet)
0.1	0.6	0.3	55.59	47.94	74.96
0.1	0.7	0.2	49.64	44.08	73.88
0.1	0.8	0.1	**69.8**	**55.0**	**82.8**
0.1	0.85	0.05	57.41	51.20	70.25

Table 1. CIFAR-10 Validation Set accuracy of the student models trained using Teacher-Student-Peer learning for different hyperparameters.

idation set. The accuracy of different hyperparameters are evaluated and results is shown in Table 1. We had proposed two methods for model selection: a) Which gives the highest accuracy for the weakest student (model which has the lowest accuracy when trained from scratch) and b) Which gives the highest accuracy for the strongest student (model which has the highest accuracy when trained from scratch). Based on both the methods, we select the hyperparameters: $\alpha = 0.1$, $\beta = 0.8$, and $\gamma = 0.1$. We continued with these hyperparameters for further experiments.

Figure 2 and 3 shows the loss and accuracy of the Teacher-Student-Peer learning vs. the vanilla Teacher-Student Learning and scratch learning. For each student, we see that the training accuracy is lower in comparison to training from scratch and Teacher-Student learning. However, we observe a good performance in the test set. In the Figure 3 and in the Table 2 we see a higher performance in test set in comparison to the training accuracy. We also see an improvement in performance in the test set of two student models in comparison to training from scratch. Particularly, a 9.26% absolute improvement for AlexNet and a 4.72% absolute improvement for MobileNet. In addition, they also improved performance in comparison to the vanilla Teacher-Student model. Particularly, a 8.28% absolute improvement for AlexNet, and a 2.13% absolute improvement for MobileNet. However, for LeNet model, we didn't observed an improvement in performance. But the performance of our approach for LeNet was comparable with learning from scratch and learning from the vanilla approach.

One observation is that peer learning is helping students which has a higher capacity and ability to take cues from other models. Also, it is promoting their ability to generalize well on test set. However, it didn't improve the performance of very low capacity students.

4.6.2 Teacher-Student-Peer Alternate learning

We experimented with an Alternate Learning scheme with our Teacher-Student-Peer approach. As described in Section 3.3, we train two students from the Teacher. For this, we selected the hyperparameters $\alpha = 0.1$, $\beta = 0.8$, and $\gamma = 0.1$, which was shown to perform better using our hyperparameter tuning experiment. However, since there are only two students, each student has only one peer model. Once the students are learned, the teacher and the learned students teach the unlearned student. For this, we had to tune our weights (hyperparameters) to give comparatively equal weightage to our peers who are now learned. We went with $\alpha = 0.1$, $\beta = 0.6$, and $\gamma = 0.3$ which showed reasonably decent performance in our hyperparameter tuning experiment.

Figure 2 and 3 shows the loss and accuracy of the Teacher-Student-Peer learning with Alternate Learning scheme vs. the Teacher-Student-Peer learning without Alternate Learning scheme, vanilla Teacher-Student Learning and scratch learning. The loss and accuracy shows the loss and accuracy of the student when they are learning from their teacher and learned peers. One clear observation is that they perform similar to Teacher-Student-Peer learning. Both have training accuracy lower than training from scratch and vanilla Teacher-Student learning. But, they show good performance in test set. For all three students, this scheme shows improved training accuracy than Teacher-Student-Peer learning. The learned peers are helping in getting higher training accuracy from the students. In test set, for two students, particularly AlexNet and MobileNet, we see an improvement over training from scratch and in one student, AlexNet, we saw improvement over the vanilla Teacher-Student model. For MobileNet, we see an 1.01% absolute improvement in accuracy from scratch learning. For AlexNet, we saw an 3.92% absolute accuracy improvement over scratch learning and an 2.94% absolute accuracy improvement over vanilla Teacher-Student approach.

A key observation from this alternate learning experiment is that when we have more peers students, it tends to perform better. As seen in Table 2, in most of our experiments when teacher taught two students vs. when teacher taught three students, we see that students were performing better when they learned from more peers.

When we compare the results of Teacher-Student-Peer with and without Alternate Learning, we see that more peers are a more important factor in improving performance of student than learning from highly learned peers. We believe this is because the students learn to better generalize when they learn from more peers.

4.7. Qualitative Evaluation of Performance

To understand what knowledge is getting distilled from the teacher and the peers to the students we did two evaluations.

Figure 4 shows test set images which were classified correctly by the high-performing teacher model and classified incorrectly by AlexNet (Student 1) when trained

Experiment	Test Accuracy (%)		
	Student 1 (AlexNet)	Student 2 (LeNet)	Student 3 (MobileNet)
Scratch	60.54	56.5	78.16
Vanilla Teacher-Student approach	61.52	**58.02**	80.75
Teacher-Student-Peer approach (Ours)	**69.8**	55.0	**82.88**
Teacher-Student-Peer Alternate Learning: Teacher to S1, S2 (Ours)	63.83	57.94	-
Teacher-Student-Peer Alternate Learning: Teacher, S1, S2 to S3 (Ours)	-	-	79.17
Teacher-Student-Peer Alternate Learning: Teacher to S1, S3 (Ours)	67.33	-	82.83
Teacher-Student-Peer Alternate Learning: Teacher, S1, S3 to S2 (Ours)	-	55.14	-
Teacher-Student-Peer Alternate Learning: Teacher to S2, S3 (Ours)	-	54.90	82.24
Teacher-Student-Peer Alternate Learning: Teacher, S2, S3 to S1 (Ours)	64.46	-	-

Table 2. Results: Comparison of learning from scratch vs. learning from Teacher-Student approach vs. learning from Teacher-Student-Peer approach with and without alternate learning

Figure 4. Images classified incorrectly by student when trained from scratch, classified correctly by teacher, and classified correctly by student taught using Teacher-Student approach.

Figure 5. Images classified incorrectly by student when trained from scratch, classified incorrectly by teacher, classified correctly by Student 2 (LeNet) and classified correctly by student taught using Teacher-Student-Peer approach.

from scratch. However, when we trained Student 1 using Teacher-Student approach, these images were correctly classified by the student. It appears that ResNet was good at identifying animals, and was able to transfer this knowledge to the student which it couldn't learn when trained from scratch.

In order to see what benefit our approach brings to the table, we saw test set images which were classified incorrectly by the high-performing teacher model and classified incorrectly by AlexNet (Student 1) when trained from scratch, but one of its peer model LeNet (Student 2) was able to correctly classify. With this setting, we saw that Student 1 trained with its peers using the Teacher-Student-Peer approach was able to correctly classify some of these images (these are the images shown in figure 5). In other words, the model is not only learning from its teacher but also learning from its peers. As shown in Figure 5, it appears that peer model LeNet (Student 2) was very good at identifying structures or lines, and was able to transfer this knowledge to the student which it couldn't learn easily when trained from scratch and when trained just by the teacher.

5. Conclusion

Our experiment suggests that our proposed approach of Teacher-Student-Peer was performing well in comparison to the vanilla Teacher-Student model. We would like to extend this experiment with more students in the ensemble. We would also like to experiment with more epochs. We believe that the students will learn more when they learn with more peers and for a longer time. Also, we would like to compare our approach against other Knowledge Distillation approaches which are currently proposed and/or being used for Knowledge Distillation.

References

[1] Alexandru Niculescu-Mizil Cristian Bucila, Rich Caruana. Model compression, 2006. 1

[2] Jianping Gou, Baosheng Yu, Stephen John Maybank, and Dacheng Tao. Knowledge distillation: A survey. *CoRR*, abs/2006.05525, 2020. 1

[3] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. 1, 4

[4] Zhe Zhao-Dong Lin Anima Singh Ed H.Chi Sagar Jain Jiaxi Tang, Rakesh Shivanna. Understanding and improving knowledge distillation, 2020. 1

[5] Kuk-Jin Yoon Lin Wang. Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks, 2020. 2

[6] Shai Shalev-Shwartz Ruth Urner, Shai Ben-David. Access to unlabeled data can speed up prediction time, 2016. 1

[7] Ang Li-Nir Levine Akihiro Matsukawa Hassan Ghasemzadeh Seyed Iman Mirzadeh, Mehrdad Farajtabar. Improved knowledge distillation via teacher assistant, 2020. 1

[8] Antonio Torralba-Alexei A. Efros Tongzhou Wang, Jun-Yan Zhu. Dataset distillation, 2018. 2

[9] Guile Wu and Shaogang Gong. Peer collaborative learning for online knowledge distillation, 2021. 1