Knowledge Distillation Doesn't Really Work

In this paper, we attempted to successfully implement knowledge distillation with MNIST. However, we found that the distilled student underperformed the student alone for all tested hyperparameters. This result was unexpected but not surprising. Many papers have reported lower accuracy after knowledge distillation. There is obviously still a gap in our understanding.

Introduction

Knowledge distillation is a model compression technique. It involves training a smaller (student) model to match the predictions of a larger (teacher) model (Hinton et al., 2015). However, it does not always work as expected (Stanton et al., 2021). This paper aims to successfully implement knowledge distillation with MNIST. We expect the distilled student to be worse than the teacher, but better than the student alone.

Background

Hinton et al. proposed knowledge distillation in "Distilling the Knowledge in a Neural Network" (2015). It compresses a larger (teacher) model into a smaller (student) model using a distillation loss function (4).

The softmax function takes a vector and returns a probability distribution. It is often the last activation function of a neural network. It is defined as

$$\sigma(y,\ t)_i = \frac{exp(y_i / t)}{\Sigma_i\, exp(y_i / t)} \tag{1}$$

where y is a vector and t is a temperature. A higher temperature returns a higher-entropy distribution (soft predictions). A lower temperature returns a lower-entropy distribution (hard predictions).

The Kullback–Leibler divergence and cross-entropy loss functions each take two vectors and return a distance. They are often the loss function of a neural network. They are defined as

$$D_{KL}(y \,||\, \bar{y}) = \Sigma_i\, y_i\, log\, \frac{y_i}{\bar{y}_i} \tag{2}$$

and

$$H(y,\ \bar{y}) = -\ \Sigma_i\, y_i\, log\, \bar{y}_i \tag{3}$$

where y (predicted labels) and $\bar{y}$ (true labels) are vectors.

Usually, the distillation loss function is defined as the weighted sum of the cross-entropy loss between the hard student predictions and the true labels, and the scaled Kullback-Leibler divergence between the soft student predictions and the soft teacher predictions

$$L(y_s,\ y_t,\ \bar{y},\ t,\ \alpha) = \alpha\, H(\sigma(y_s,\ 1),\ \bar{y}) + (1 - \alpha)\, D_{KL}(\sigma(y_s,\ t) \,||\, \sigma(y_t,\ t))\, t^2 \quad (4)$$

where y_s (student predictions) and y_t (teacher predictions) are vectors, and α is a weight in [0, 1].

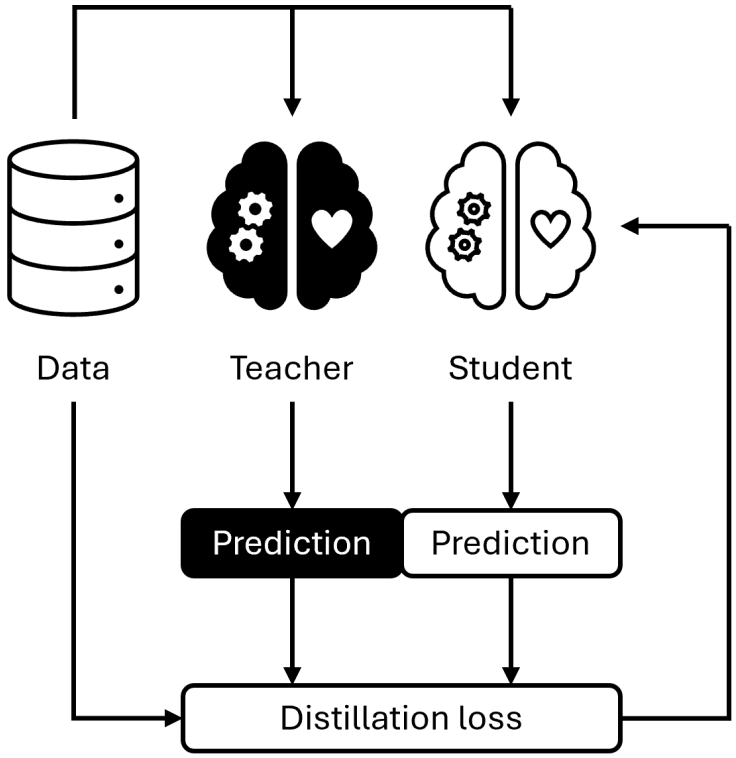

Figure 1. Knowledge Distillation. A teacher and student each make a prediction. The distillation loss is calculated using the teacher prediction, student prediction, and true label. The student is updated.

Consequently, both the student and teacher make predictions. However, only the student is updated (Figure 1).

Methods

MNIST was loaded (Figure 2). The images were scaled to [0, 1]. It was split into stratified train (50,000), test (10,000), and validation (10,000) subsets.

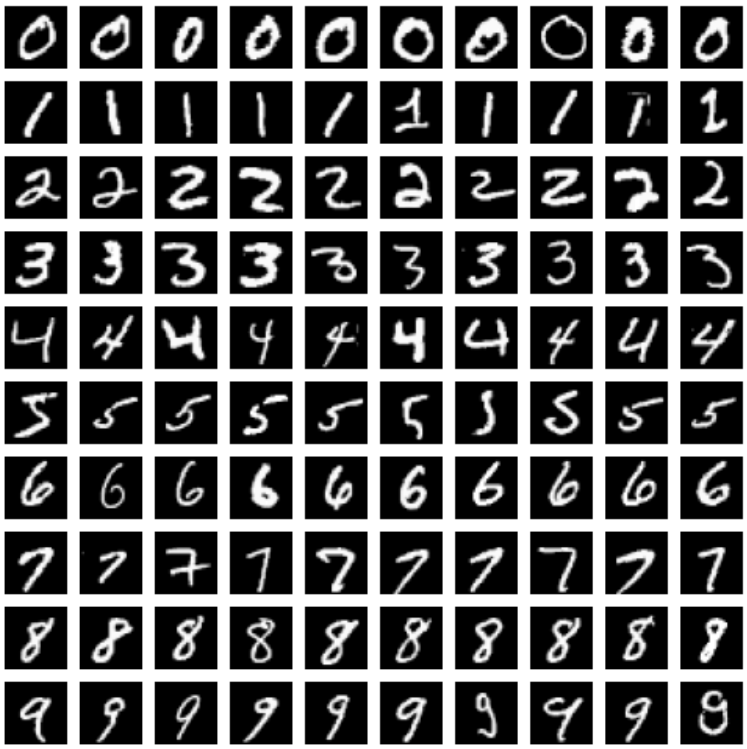

Figure 2. MNIST. A set of 70,000 28x28 handwritten digits 0-9.

A teacher was created (Table 1). It was trained for 20 epochs and frozen.

layer	output	parameters
Input	(None, 28, 28, 1)	0
Conv2D	(None, 28, 28, 8)	80
MaxPooling2D	(None, 14, 14, 8)	0
Conv2D	(None, 14, 14, 16)	1168
MaxPooling2D	(None, 7, 7, 16)	0
Flatten	(None, 784)	0
Dense	(None, 32)	25120
Dropout	(None, 32)	0
Dense	(None, 10)	330
total		26698

Table 1. Teacher Architecture. It takes a 28x28 image and returns a 10D prediction. It has 4 trainable layers: a convolutional layer with 8 3x3 filters and ReLU activation, a convolutional layer with 16 3x3 filters and ReLU activation, a fully-connected layer with 32 units and ReLU activation, and a fully-connected layer with 10 units.

A student was created (Table 2). It was cloned and trained for 20 epochs.

layer	output	parameters
Input	(None, 28, 28, 1)	0
Flatten	(None, 784)	0
Dense	(None, 32)	25120
Dropout	(None, 32)	0
Dense	(None, 10)	330
total		25450

Table 2. Student Architecture. It takes a 28x28 image and returns a 10D prediction. It has 2 trainable layers: a fully-connected layer with 32 units and ReLU activation, and a fully-connected layer with 10 units.

Other clones were distilled for temperatures in [1, 10] and alphas in [0.0, 1.0] for 20 epochs.

Results

The teacher and student achieved good fits (Figure 3).

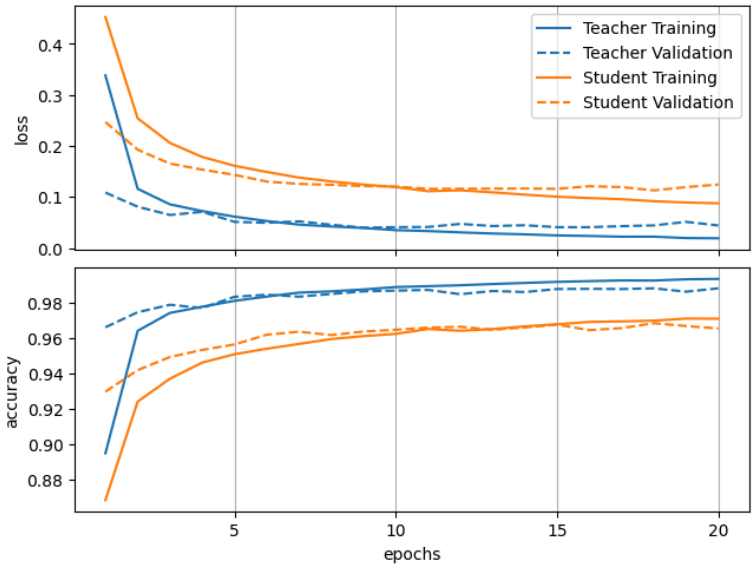

Figure 3. Teacher and Student Training and Validation Curves. The teacher achieved a good fit after 3 epochs and overfit after 11 epochs. The student achieved a good fit after 7 epochs and overfit after 13 epochs.

They achieved accuracies of 98.4% and 96.3% (Table 3).

label	teacher			student			support
	precision	recall	f1-score	precision	recall	f1-score	
Zero	0.99	0.99	0.99	0.99	0.97	0.98	986
One	0.99	0.99	0.99	0.98	0.98	0.98	1125
Two	0.99	0.98	0.98	0.96	0.96	0.96	999
Three	0.98	0.99	0.99	0.96	0.95	0.96	1020
Four	0.99	0.98	0.99	0.95	0.97	0.96	975
Five	0.97	0.98	0.98	0.94	0.95	0.95	902
Six	0.99	0.99	0.99	0.96	0.98	0.97	982
Seven	0.97	1.00	0.98	0.97	0.98	0.97	1042
Eight	0.99	0.97	0.98	0.96	0.94	0.95	975
Nine	0.98	0.97	0.98	0.96	0.95	0.96	994
accuracy			0.98			0.96	10000
macro avg	0.98	0.98	0.98	0.96	0.96	0.96	10000
weighted avg	0.98	0.98	0.98	0.96	0.96	0.96	10000

Table 3. Teacher and Student Classification Report. The teacher and student achieved accuracies of 98.4% and 96.3%.

They had similar confusion patterns (Figure 4).

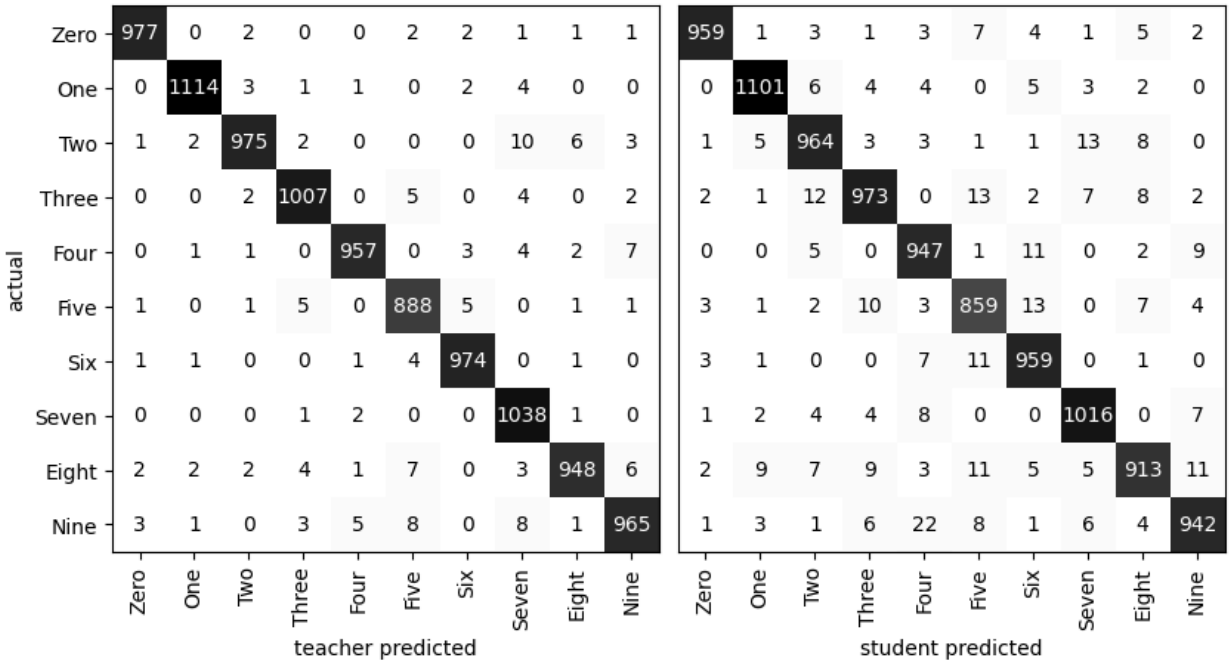

Figure 4. Teacher and Student Confusion Matrices. The teacher and student often confused nines and fours, fives and threes, and sixes and fives.

The distilled students achieved accuracies between 95.7% and 96.8%. Accuracy remained constant when temperature or alpha was 1. Otherwise, accuracy decreased as temperature increased or alpha decreased (Figure 5).

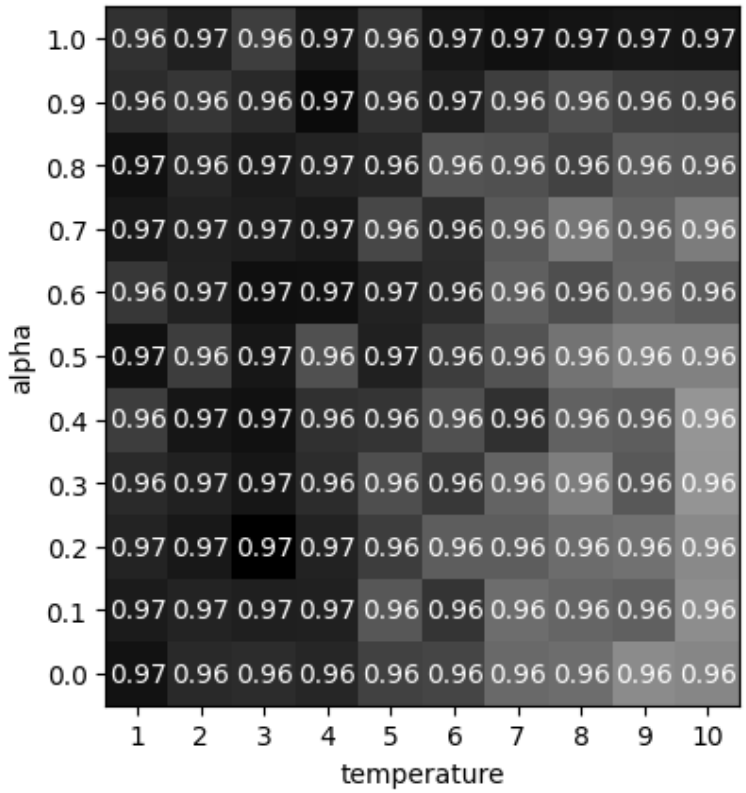

Figure 5. Knowledge Distillation Grid Search. Accuracy remained constant when temperature or alpha was 1. Otherwise, accuracy decreased as temperature increased or alpha decreased.

Discussion

The teacher was a convolutional neural network with 26,698 layers. The student was derived from the teacher by removing the convolutional and max pooling layers. Consequently, the student was a neural network with 25,450 parameters (Table 1; Table 2). This setup was nearly a self-distillation. However, the teacher was still compressed.

Both the teacher and student achieved good fits (Figure 3) and showed similar confusion patterns (Figure 4). Additionally, the student accuracy was less than the teacher accuracy (Table 3). We expected the accuracies of the distilled students to be greater than the student alone.

During the grid search, accuracy remained constant when temperature or alpha was 1 (Figure 5). This makes sense because, when temperature is 1, the shape of the distillation loss function is the same for all alphas (assuming a perfect teacher). Similarly, when alpha is 1, the argmax of the predictions is the same for all temperatures.

However, accuracy decreased as temperature increased or alpha decreased (Figure 5). Ultimately, the alpha of the best distilled students were 1, making them identical to the student alone.

Possible explanations include underfitting, overfitting, and a bad search space. Underfitting is unlikely since the student achieved a good fit on the true labels (Figure 3) and the teacher is (almost) perfect (Table 3). Overfitting is also unlikely since the accuracies of the distilled students were worse than the student alone (Table 3; Figure 5). A bad search space is possible if the optimal temperature is greater than 10.

While this result was unexpected, it was not surprising. Many papers report lower accuracy after knowledge distillation (Cho & Hariharan, 2019; Borup, 2020; Sayak, 2021; Chariton, 2024). While we expected the conditions in this paper to be ideal, there is obviously still a gap in our understanding.

Conclusion

This paper aimed to successfully implement knowledge distillation with MNIST. We expected the distilled student to be worse than the teacher, but better than the student alone. However, we found that the distilled student was worse than the student alone. Many papers have reported similar conclusions. Nonetheless, knowledge distillation remains an active area of research. Knowledge distillation does really work, but it is difficult to achieve. We look forward to trying new knowledge distillation techniques (Park et al., 2019; Tung & Mori, 2019; Mirzadeh et al., 202; Zhao et al., 2022; Wei & Bai, 2024).

Acknowledgements

We would like to thank Specs for letting us use his GPU.

References

Borup, K. (2020). Knowledge Distillation. In Keras. Retrieved August 6, 2024, from https://keras.io/examples/vision/knowledge_distillation/

Chariton, A. (2024). Knowledge Distillation Tutorial. Retrieved August 6, 2024 from https://pytorch.org/tutorials/beginner/knowledge_distillation_tutorial.html

Cho, J. H., & Hariharan, B. (2019). On the Efficacy of Knowledge Distillation. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 4794-4802).

Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531*.

Mirzadeh, S. I., Farajtabar, M., Li, A., Levine, N., Matsukawa, A., & Ghasemzadeh, H. (2020, April). Improved Knowledge Distillation via Teacher Assistant. *In Proceedings of the AAAI conference on artificial intelligence* (Vol. 34, No. 04, pp. 5191-5198).

Park, W., Kim, D., Lu, Y., & Cho, M. (2019). Relational Knowledge Distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 3967-3976).

Sayak, P. (2021). Knowledge Distillation Recipes. In *Keras*. Retrieved August 6, 2024, from https://keras.io/examples/keras_recipes/better_knowledge_distillation/

Stanton, S., Izmailov, P., Kirichenko, P., Alemi, A. A., & Wilson, A. G. (2021). Does Knowledge Distillation Really Work?. *Advances in Neural Information Processing Systems, 34*, 6906-6919.

Tung, F., & Mori, G. (2019). Similarity-preserving Knowledge Distillation. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 1365-1374).

Wei, Y., & Bai, Y. (2024). Dynamic Temperature Knowledge Distillation. *arXiv preprint arXiv:2404.12711*.

Zhao, B., Cui, Q., Song, R., Qiu, Y., & Liang, J. (2022). Decoupled Knowledge Distillation. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition* (pp. 11953-11962).