Does Knowledge Distillation Really Work?

Samuel Stanton
NYU

Pavel Izmailov
NYU

Polina Kirichenko
NYU

Alexander A. Alemi
Google Research

Andrew Gordon Wilson
NYU

Abstract

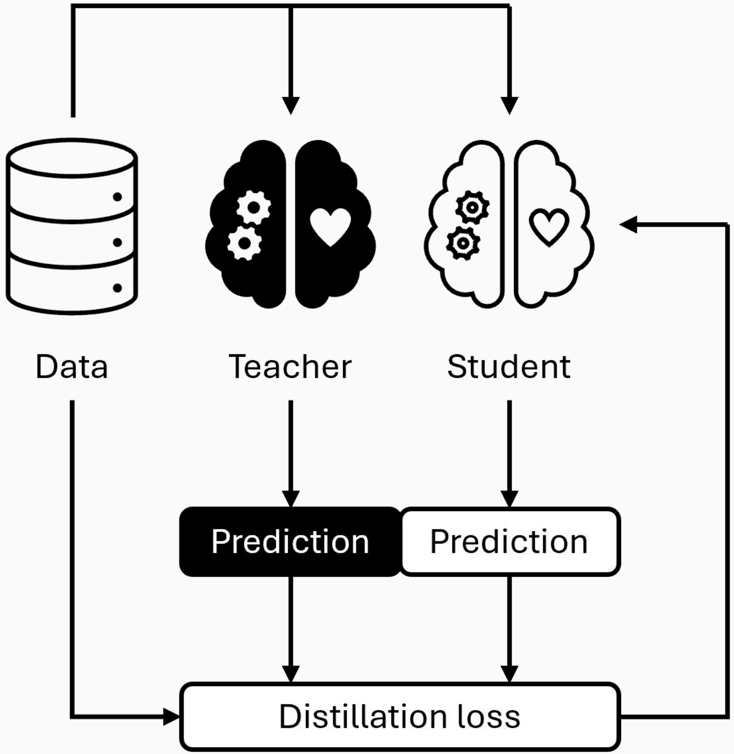
Knowledge distillation is a popular technique for training a small student network to emulate a larger teacher model, such as an ensemble of networks. We show that while knowledge distillation can improve student generalization, it does not typically work as it is commonly understood: there often remains a surprisingly large discrepancy between the predictive distributions of the teacher and the student,

Knowledge Distillation Doesn’t Really Work

Knowledge Distillation Doesn’t Really Work

yet…

Introduction



Knowledge distillation is a model compression technique.

It involves training a smaller (student) model to match the predictions of a larger (teacher) model.

This project aimed to successfully implement knowledge distillation with MNIST.

I expected the distilled student to be worse than the teacher, but better than the student alone.

Background (1 of 2)

The softmax function takes a vector and returns a probability distribution.

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

The Kullback–Leibler divergence and cross-entropy loss functions each take two vectors and return a distance.

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

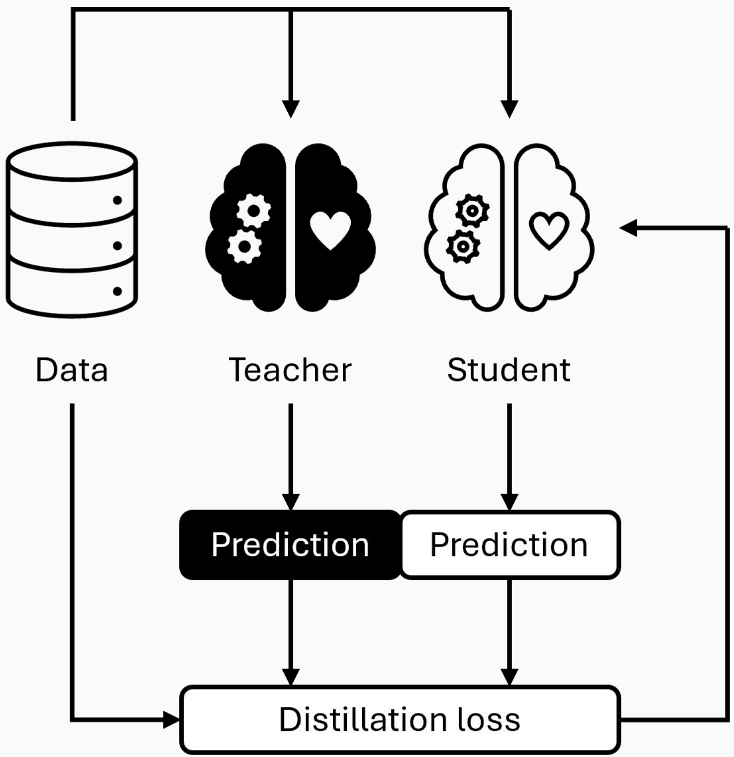
$$H(y, \bar{y}) = - \Sigma_i \, y_i \, log \, \bar{y}_i$$

The distillation loss function

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

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.

Background (2 of 2)



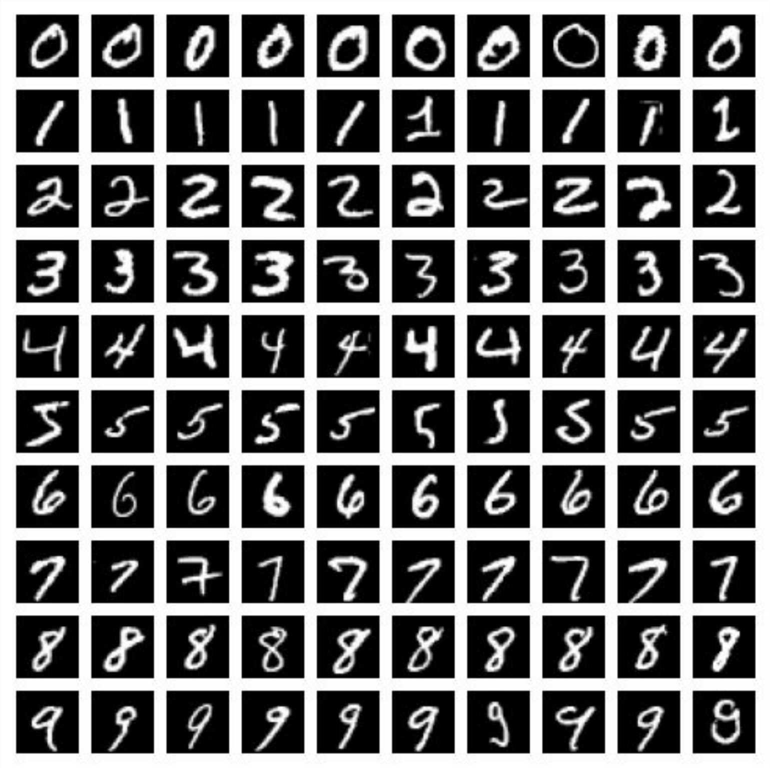
A teacher and student each make a prediction.

The distillation loss is calculated using the teacher prediction, student prediction, and true label.

The student is updated.

Both the student and teacher make predictions, but only the student is updated.

Methods (1 of 3)



1. Load MNIST.
2. Scale the images [0, 1].
3. Split into train, test, and validation.
4. Create and train the teacher.
5. Create and train the student.
6. Knowledge distillation grid search.

Methods (2 of 3)

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

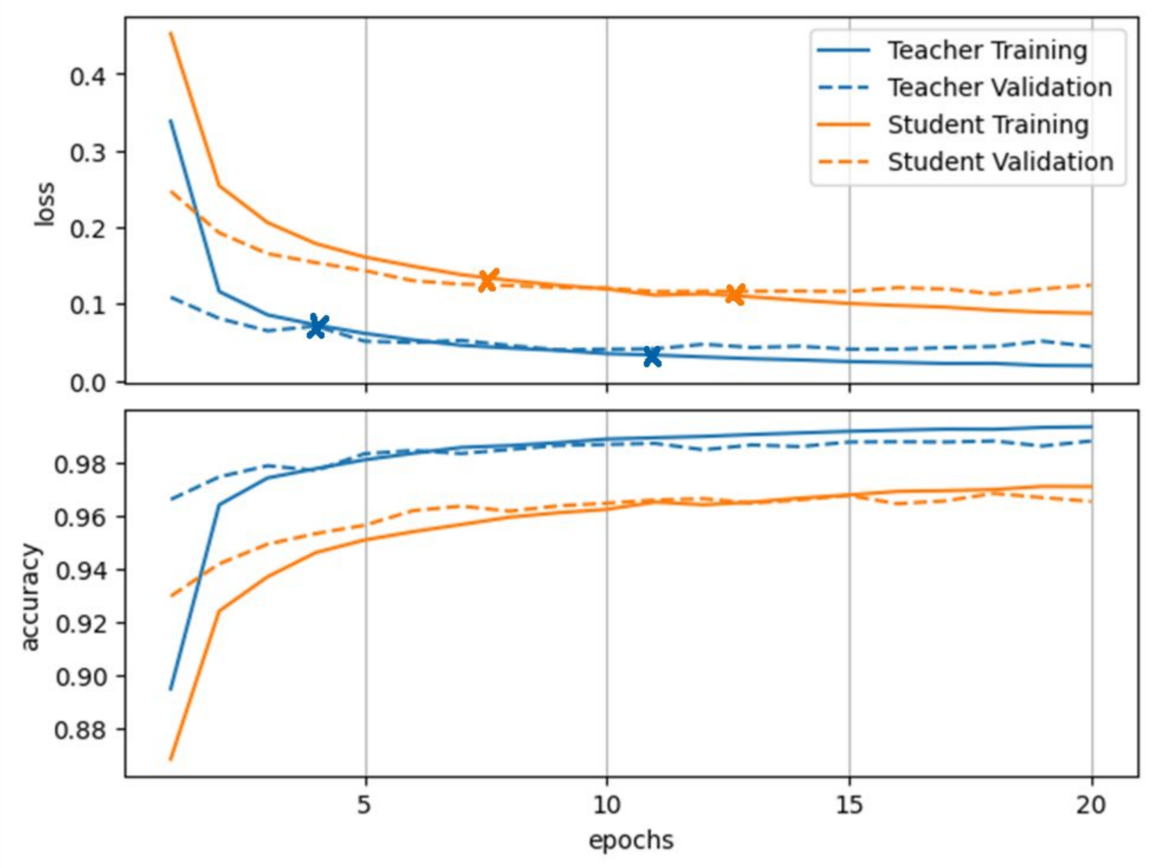
1. Load MNIST.
2. Scale the images [0, 1].
3. Split into train, test, and validation.
4. Create and train the teacher.
5. Create and train the student.
6. Knowledge distillation grid search.

Methods (3 of 3)

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		25450 ~~26698~~

1. Load MNIST.
2. Scale the images [0, 1].
3. Split into train, test, and validation.
4. Create and train the teacher.
5. Create and train the student.
6. Knowledge distillation grid search.

Results (1 of 4)



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.

Overall, both look good.

Results (2 of 4)

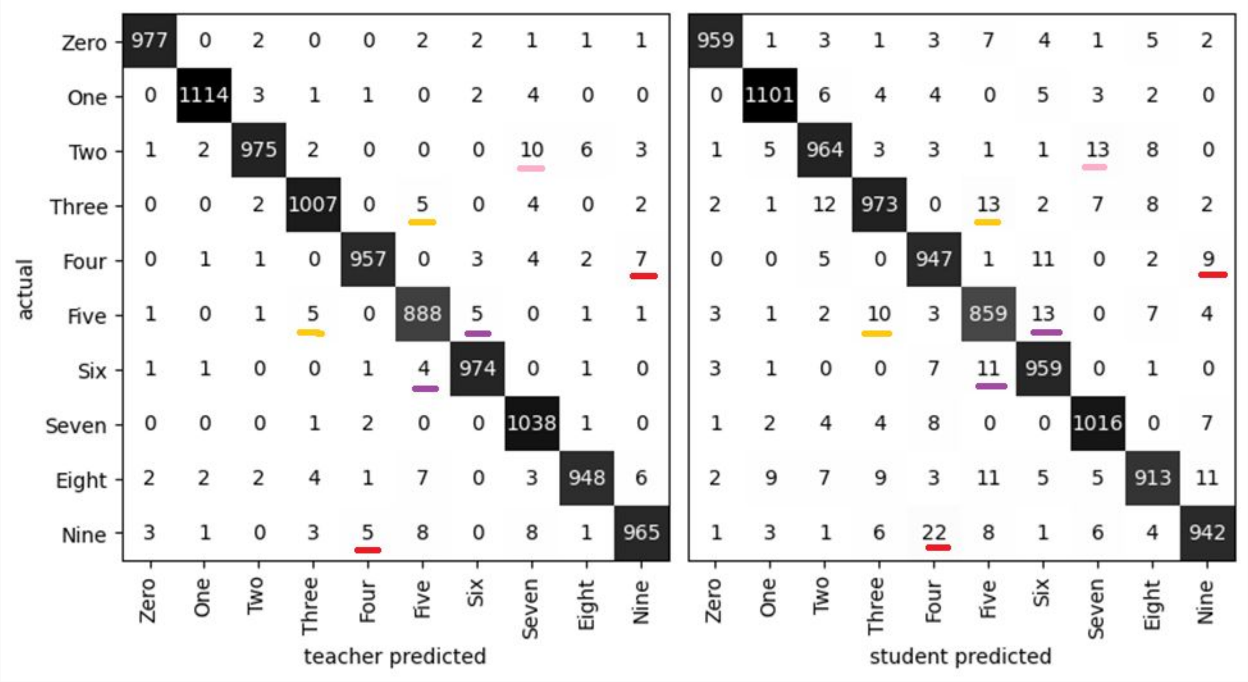
label	teacher			student			support
	precision	recall	f1-score	precision	recall	f1-score	
Zero	0.99	0.99	0.99	0.99	0.97	0.98	986
One	0.99	0.99	0.99	0.98	0.98	0.98	1125
Two	0.99	0.98	0.98	0.96	0.96	0.96	999
Three	0.98	0.99	0.99	0.96	0.95	0.96	1020
Four	0.99	0.98	0.99	0.95	0.97	0.96	975
Five	0.97	0.98	0.98	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

Teacher and Student Classification Report.

The teacher and student achieved accuracies of 98.4% and 96.3%.

Overall, they perform similarly, but the student is a little worse.

Results (3 of 4)

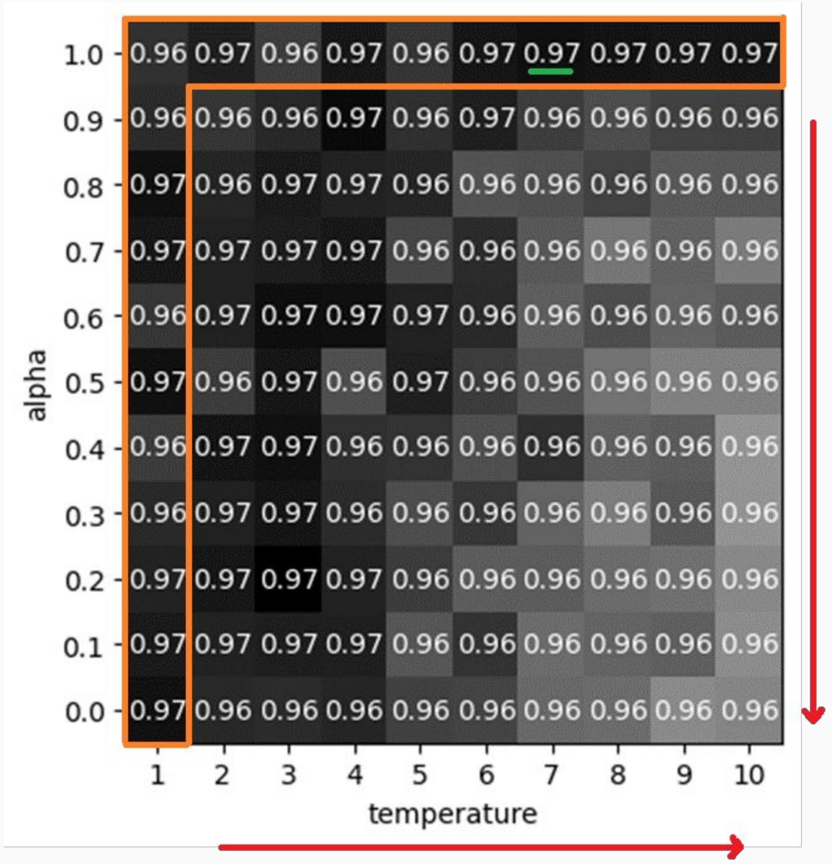


Teacher and Student Confusion Matrices.

The teacher and student often confused nines and fours, fives and threes, and sixes and fives.

Overall, both have similar confusion patterns.

Results (4 of 4)



Knowledge Distillation Grid Search.

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.

This is not what I expected.

Discussion (1 of 5)

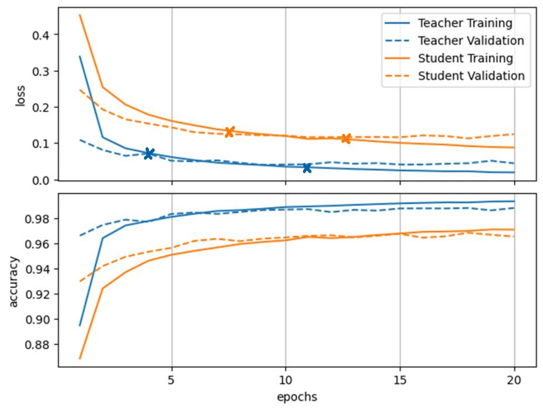
layer	output	parameters
Input	(None, 28, 28, 1)	0
~~Conv2D~~	~~(None, 28, 28, 8)~~	~~80~~
~~MaxPooling2D~~	~~(None, 14, 14, 8)~~	~~0~~
~~Conv2D~~	~~(None, 14, 14, 16)~~	~~1168~~
~~MaxPooling2D~~	~~(None, 7, 7, 16)~~	~~0~~
Flatten	(None, 784)	0
Dense	(None, 32)	25120
Dropout	(None, 32)	0
Dense	(None, 10)	330
total		25450 ~~26698~~

The student was derived from the teacher by removing the convolutional and max pooling layers.

This setup was nearly a self-distillation. However, the teacher was still compressed.

This is perfect for knowledge distillation.

Discussion (2 of 5)

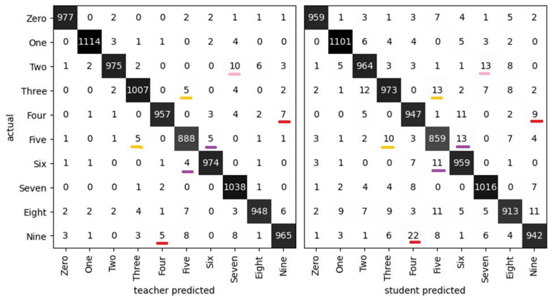


label	teacher			student			support
	precision	recall	f1-score	precision	recall	f1-score	
Zero	0.99	0.99	0.99	0.99	0.97	0.98	986
One	0.99	0.99	0.99	0.98	0.98	0.98	1125
Two	0.99	0.98	0.98	0.96	0.96	0.96	999
Three	0.98	0.99	0.99	0.96	0.95	0.96	1020
Four	0.99	0.98	0.99	0.95	0.97	0.96	975
Five	0.97	0.98	0.98	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

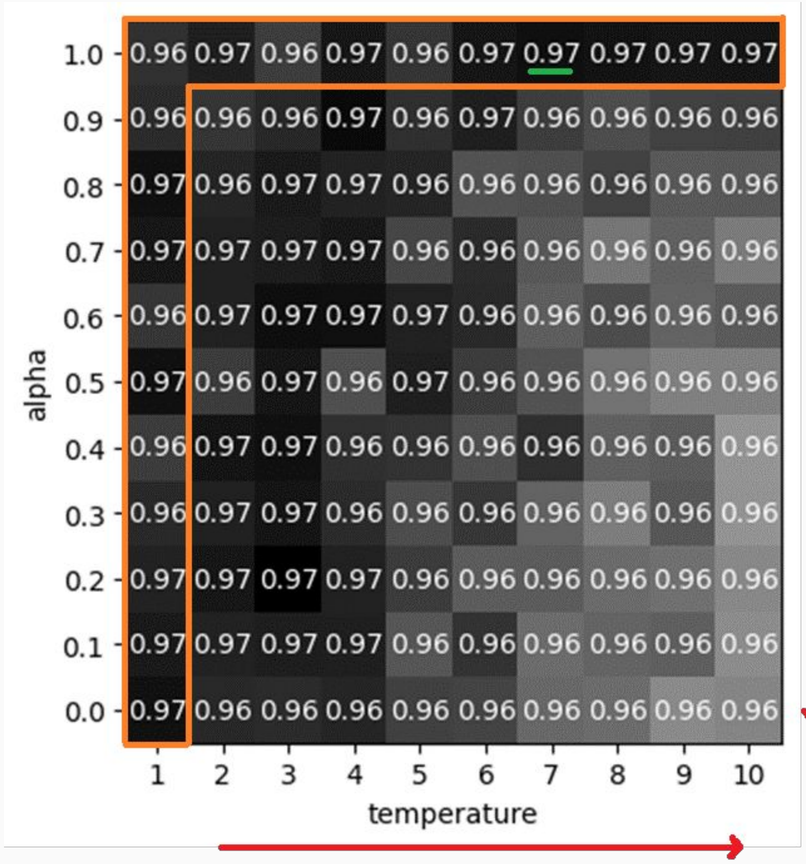
Both the teacher and student achieved good fits and showed similar confusion patterns.

Additionally, the student accuracy was less than the teacher accuracy.

This is perfect for knowledge distillation.



Discussion (3 of 5)



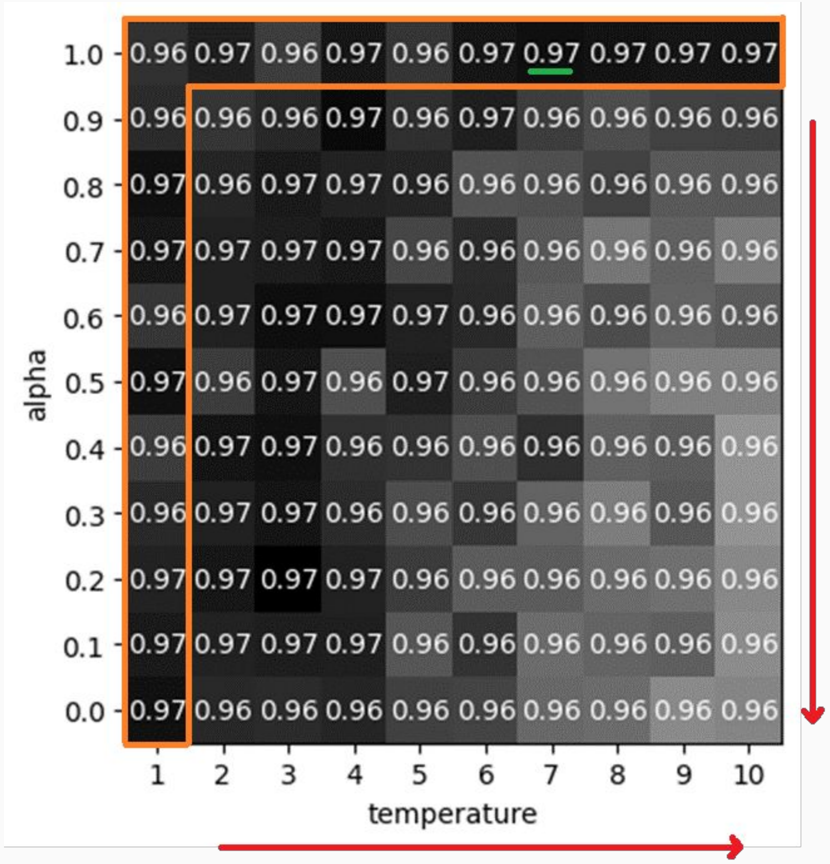
During the grid search, accuracy remained constant when temperature or alpha was 1.

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).

When alpha is 1, the argmax of the predictions is the same for all temperatures.

Discussion (4 of 5)

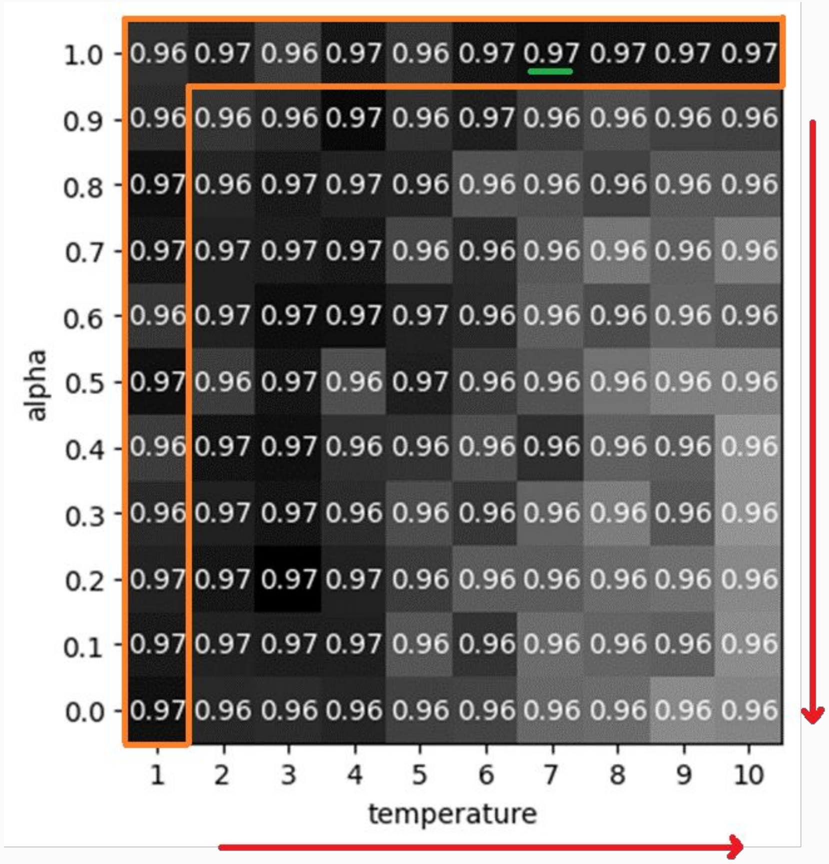


However, accuracy decreased as temperature increased or alpha decreased.

The alpha of the best distilled students were 1, making them identical to the student alone.

What happened?

Discussion (5 of 5)



Underfitting is unlikely since the student achieved a good fit on the true labels and the teacher is (almost) perfect.

Overfitting is also unlikely since the accuracies of the distilled students were worse than the student alone.

A bad search space is possible if the optimal temperature is greater than 10.

Overall, I’m not sure why this happened.

Conclusion

This project aimed to successfully implement knowledge distillation with MNIST.

I expected the distilled student to be worse than the teacher, but better than the student alone. However, I found that the distilled student was worse than the student alone.

Many papers have reported similar conclusions. **Knowledge distillation does really work, but it is difficult to achieve**. There is obviously still a gap in our understanding.

Nonetheless, knowledge distillation remains an active area of research. I look forward to trying new knowledge distillation techniques.

```
# Evaluate student on test dataset
distiller.evaluate(x_test, y_test)
```

```
[0.017046602442860603, 0.969200074672699]
```

```
# Train and evaluate student trained from scratch.
student_scratch.fit(x_train, y_train, epochs=3)
student_scratch.evaluate(x_test, y_test)
```

```
[0.0629437193274498, 0.9778000712394714]
```

Method	Teacher	Top-1 Error (%)
Scratch	-	30.24
Full KD [12]	ResNet18	30.57
Full KD [12]	ResNet34	30.79
Full KD [12]	ResNet50	30.95

```
_, top1_accuracy = pretrained_student.evaluate(test_ds)
print(f"Top-1 accuracy on the test set: {round(top1_accuracy * 100, 2)}%")
```

```
97/97 [==============================]
Top-1 accuracy on the test set: 81.02%
```

```
student = distiller.student
student_model.compile(metrics=["accuracy"])
_, top1_accuracy = student.evaluate(test_ds)
print(f"Top-1 accuracy on the test set: {round(top1_accuracy * 100, 2)}%")
```

```
97/97 [==============================]
Top-1 accuracy on the test set: 1.07%
```

References (1 of 2)

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).

References (2 of 2)

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).