# Reproducing Knowledge Distillation on MNIST

Andrei Chubarau

February 20, 2025

## 1 Introduction

In this document, we present a reproduction study of knowledge distillation (KD) proposed by Hinton et al. [1]. Knowledge distillation is a training technique where a smaller student model is trained to mimic the outputs of a larger teacher model. The knowledge from a teacher model is thus "distilled" into the student, with the goal of improved final performance for the student network. We reproduce the experiments by Hinton et al. on MNIST, training neural networks with and without KD. We further verify the effect of the *temperature* parameter in KD across several student network configurations.

## 2 Methodology

Following Hinton et al. [2, 1], the teacher model is a fully connected network with two hidden layers each with 1200 units with ReLU activations. The final layer of the model predicts 10 values representing class *probabilities* of each of the 10 possible digits in MNIST. The network is regularized with dropout (input and hidden layers use 0.2 and 0.5 dropout rates, respectively) and weight decay. When training the teacher model, input images are further randomly jittered up to 2 pixels as additional regularization. Following the implementation details presented in Section 3 of [1], we disable all regularization when training the student models. We train models on MNIST by optimizing the standard cross-entropy loss between the predicted class logits and the expected class labels. For KD, in addition to cross-entropy loss, we optimize Kullback-Leibler (KL) divergence between the logits predicted by the teacher and the student networks. We further optionally omit the digit 3 from the training set when training the student to test the effect of knowledge transfer on unseen data.

Where possible, we follow hyperparameters used by Hinton et al. [2, 1]. To reduce compute, we train for fewer epochs. Hinton et al. train for 3000 epochs, which is excessive—we find that the error on the test set is relatively stable after 25 epochs of training. In all our tests, we therefore train for 25 epochs. We use the SGD optimizer

with momentum 0.9, starting learning rate 0.01, which is exponentially decayed by factor of 0.95 after each epoch, and weight decay of 0.0001. For reproducibility, we use the same random seed for all train runs.

In our initial tests, we also found that if input images are randomly jittered by up to 2 pixels (as done by Hinton et al.), the effect of dropout is detrimental to the teacher's performance at our compute budget (25 training epochs). When training with random jitter for 25 epochs, the teacher model with no dropout reaches a lower error rate on the test set of MNIST, with a final performance of roughly 120 errors, while the model with dropout reaches roughly 140 errors. To remain consistent with Hinton et al., we still use both jitter and dropout for the teacher. We additionally clip gradient norm to stabilize training as we observe larger loss values when using KD (in particular for higher temperature values).

# 3 Experiments and Results

Hinton et al. first demonstrate the effect of KD by training a student network with a hidden dimension of 800 at a temperature of 20. Our results for this experiment are shown in Figure 1. The teacher network reaches 143 errors while the student yields 188 errors; the distilled student produces 158 errors, showcasing the benefit of KD[1]. We proceed to test KD under different configurations, as presented in Figure 2. We train models with hidden dimension $D \in \{30, 300, 800\}$ and apply KD with a temperature parameter $T \in \{1, 2, 4, 10, 20, 50\}$. As in Hinton et al., we find that KD is sensitive to the temperature parameter, with some values leading to stronger results. For instance, for a student with 30 hidden units, Hinton et al. observe that temperatures in the range 2.5-4 produce best results, while for students with more than 300 hidden neurons, temperatures

[1]Our error is generally higher than the error reported by Hinton et al. due to running fewer training epochs.

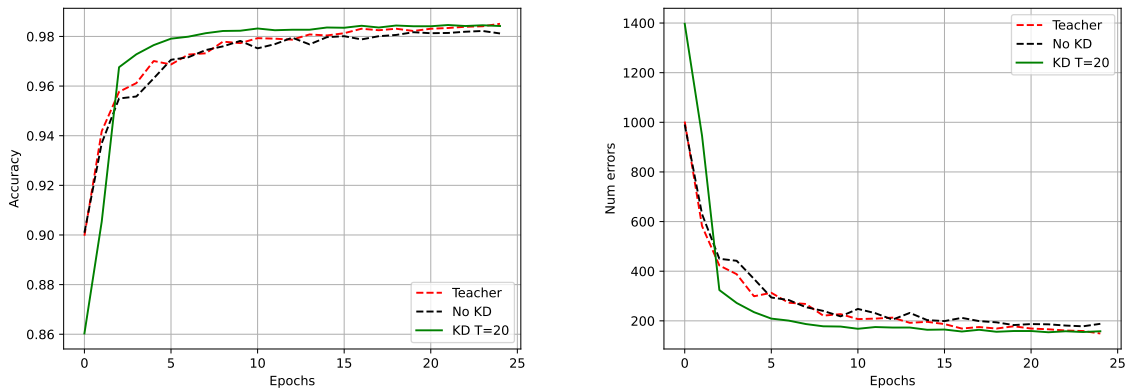
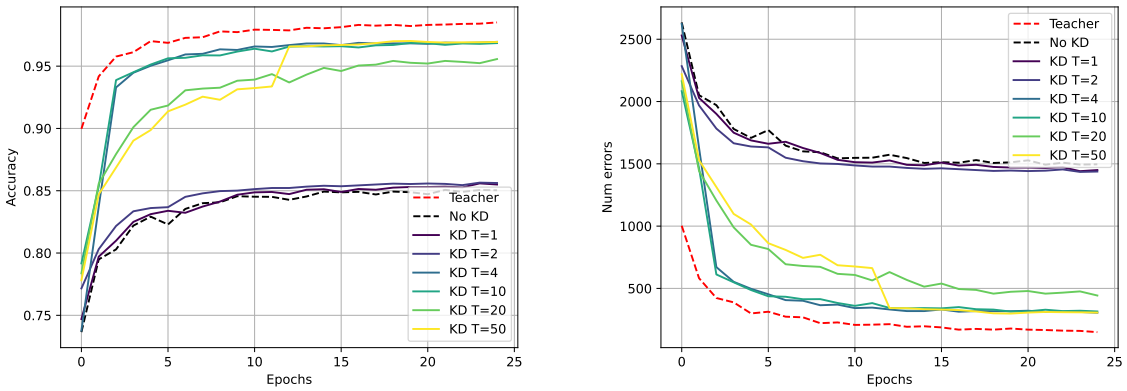
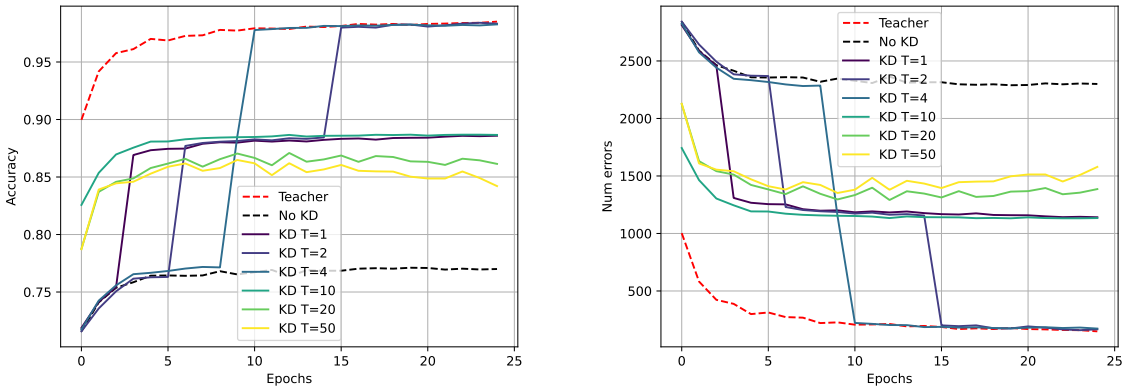


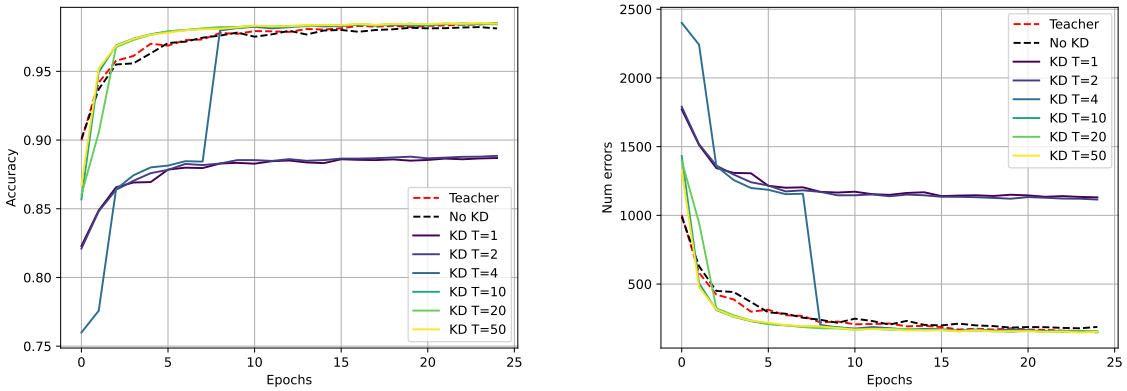

Figure 1: Effect of knowledge distillation.




(a) Student with 30 hidden units




(b) Student with 300 hidden units




(c) Student with 800 hidden units

Figure 2: Model performance on the test set of MNIST for different model configurations and temperature values used during knowledge distillation.

above 20 produce similar improvements. Our results indicate that higher temperatures tend to produce better results for all tested model configurations. We observe some inconsistencies for a student with 300 neurons, which we attribute to SGD getting stuck at and potentially escaping local minima.

We further evaluate KD while omitting the digit "3" from the transfer set, such that the student network never sees digit 3 during training. Our results confirm that, as reported by Hinton et al., the student network trained with KD produces significantly fewer errors when classifying images of the unseen digit 3 compared to a network trained without distillation. Specifically, we find that the student makes a total of 224 errors of which 100 are on the 1010 threes in the test set. A network trained similarly while omitting threes but without distillation has near zero accuracy on the threes in the test set (it missclassifies nearly all 1010 threes in the test set). Note that Hinton et al. report 206 errors on the test set (133/1010 threes) for the student model. This result supports the hypothesis that KD "can transfer a great deal of knowledge to the distilled model" [1], even for unseen data, which motivates its applications to training smaller and more efficient models given pre-trained larger teacher networks.

# 4 Conclusion

Knowledge distillation effectively improves the performance of a smaller student network by transferring knowledge from the larger teacher network. In this report, we evaluated knowledge distillation for different temperatures and student model sizes. The results presented here align with the findings reported by Hinton et al. [1], although the accuracy of our experiments would benefit from longer training sessions (number of epochs) and better tuning of the involved hyperparameters (learning rate schedules, etc.).

# References

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

[2] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors, 2012.