CS 6073 Homework 2: Convolutional Neural Network

Luginbuhl, Dale
luginbdr@mail.uc.edu

September 30th 2023

1 Data

1.1 How many data samples are included in the dataset?

There are 70,000 samples included in the dataset. There are ten possible classses for the label, one corresponding to each digit. The total data set has 7,000 samples for each class.

1.2 Which problem will this dataset try to address?

The goal is to classify an image of a handwritten digit into one of ten classes. The classes correspond to the integers in the range $[0, 9]$.

1.3 What is the minimum value and the maximum value in the dataset?

The the input of each samlple is a single-channel monochrome image. The pixel values are in the range $[0, 255]$. The labels are in the range $[0, 9]$.

1.4 What is the dimension of each data sample?

Each data sample is a 28x28 pixel monochrome image with a single channel. Each data sample therefore has 784 dimensions.

(i.e., pixels don't simply have a value of black and white, but a level of greyness from 0 to 255) as a result of the anti-aliasing technique used by the normalization algorithm

1.5 Does the dataset have any missing information? E.g., missing features.

The dataset does not have any missing information.

1.6 What is the label of this dataset?

The label of the dataset is an integer in the range $[0, 9]$ representing the drawn digit.

1.7 How many percent of data will you use for training, validation and testing?

The dataset is pre-split into 60,000 training samples and 10,000 testing samples. All of the samples in the test set were drawn by different indivduals than the training set. There are 6,000 samples per class in the training set, and 1,000 samples per class in the training set.

We will further split the training set into a training set and a validation set. The training set will consist of 50,000 samples and the validation set will consist of 10,000 samples.

The resulting percentage splits will then be:

- Training : 71.4%
- Validation : 14.3%
- Test : 14.3%

1.8 What kind of data pre-processing will you use for your training dataset?

For each data sample, we will normalize the input, and one-hot encode the label.

2 Model

We will experiment with three different models. First a multilayer perceptron with 2 hidden layers of width 32 and 16, and an output layer of width 10 (dnn). Second we implement LeNet-5 (lenet). Third we consider ResNet18 (resnet).

Model	Accuracy
dnn	96.41
lenet	99.13
resnet	99.14

Table 1: Accuracy of highest performing models

3 Objective

Cross-entropy.

4 Optimization

We chose the Adam optimizer to take advantage of momentum in our optimization. Adam allows us to converge to an optimum value more quickly, and it allows provides the potential to escape from local minima.

5 Model Selection

Model	LR: 0.1	LR: 0.01	LR: 0.001	LR: 0.0001
dnn	9.81 & 0.12	92.71 & 1.00	96.41 & 1.00	86.31 & 0.81
lenet	9.74 & 0.06	88.06 & 1.00	99.13 & 1.00	97.86 & 1.00
resnet	9.80 & 0.06	98.06 & 1.00	99.11 & 1.00	99.14 & 1.00

Table 2: Impact of learning rate selection on model accuracy and F1 scores

In all of our models we achieved the best training results with a learning rate of 0.001 or 0.0001. Learning rates of 0.1 and 0.01 were too large for the training to be stable. For the dnn and lenet model the best results were achieved at a learning rate of 0.001. A learning rate of 0.0001 had the potential for superior results, but due to the small learning rate size, the training required too much time. For the resnet model we achieved the best results with a learning rate of 0.0001. Our results indicate that perhaps using a learning rate scheduler could have yielded superior results. It would allow us to have the benefits of a small learning rate, especially towards the end of training, but without requiring too much time to train.

We didn't have much issue with underfitting or overfitting in our models. If we needed to combat overfitting we could apply regularization, dropout, or data augmentation.

6 Model Performance

The best dnn model was the model trained with a learning rate of 0.001. The best lenet model was the model trained with a learning rate of 0.001. The best resnet model was the model trained with a learning rate of 0.0001. The best model overall was the resnet, although it edged out the lenet model by a very slim margin. With more fine-tuning, perhaps one of the models would emerge as the clear winner.

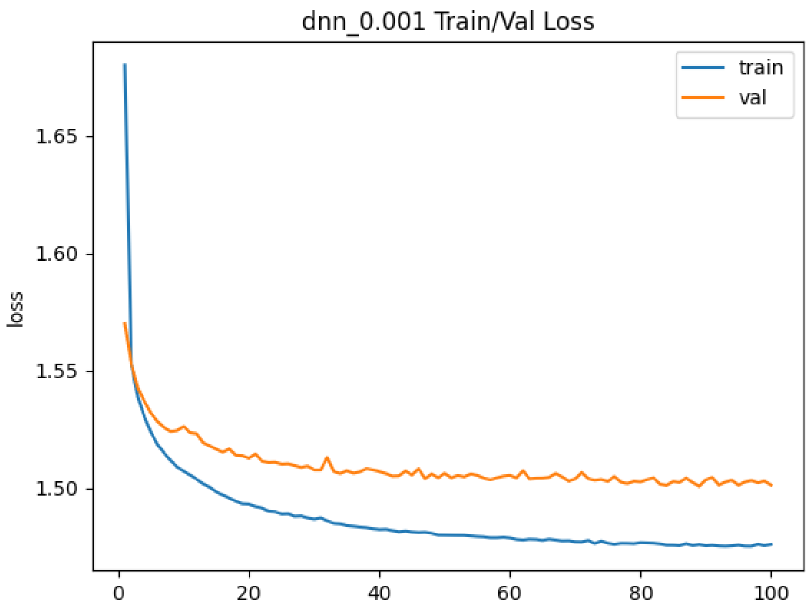

Figure 1: Training and validation loss for dnn model with a learning rate of 0.001

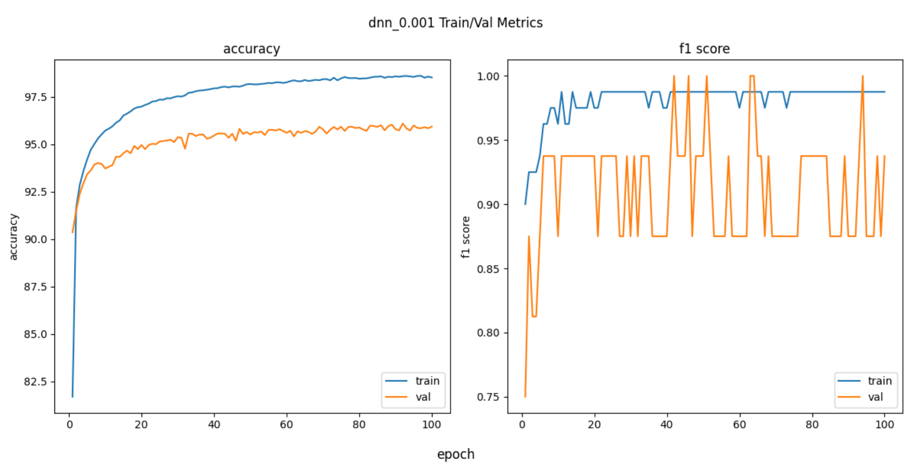

Figure 2: Training and validation accuracy and F1 scores for dnn model with a learning rate of 0.001

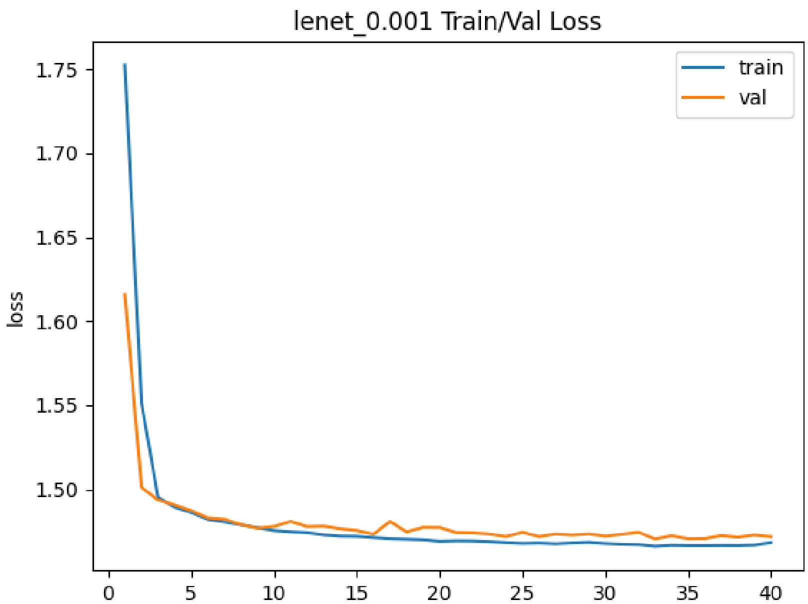

Figure 3: Training and validation loss for lenet model with a learning rate of 0.001

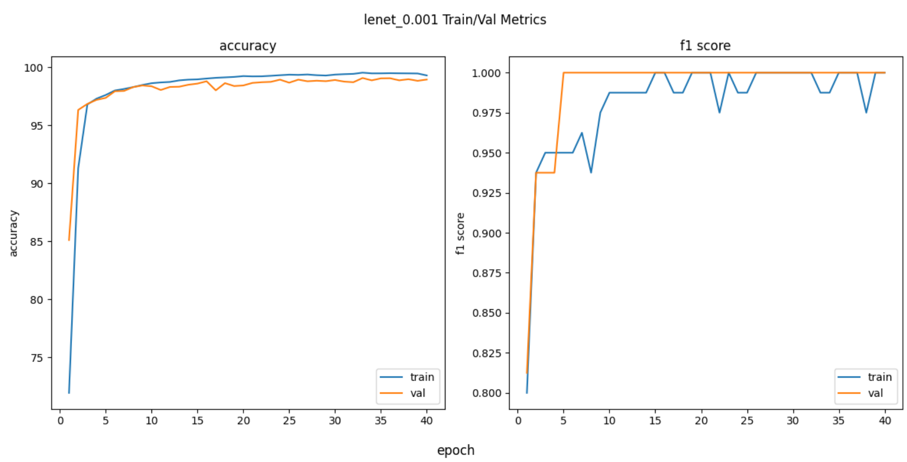

Figure 4: Training and validation accuracy and F1 scores for lenet model with a learning rate of 0.001

Figure 5: Training and validation loss for resnet model with a learning rate of 0.0001

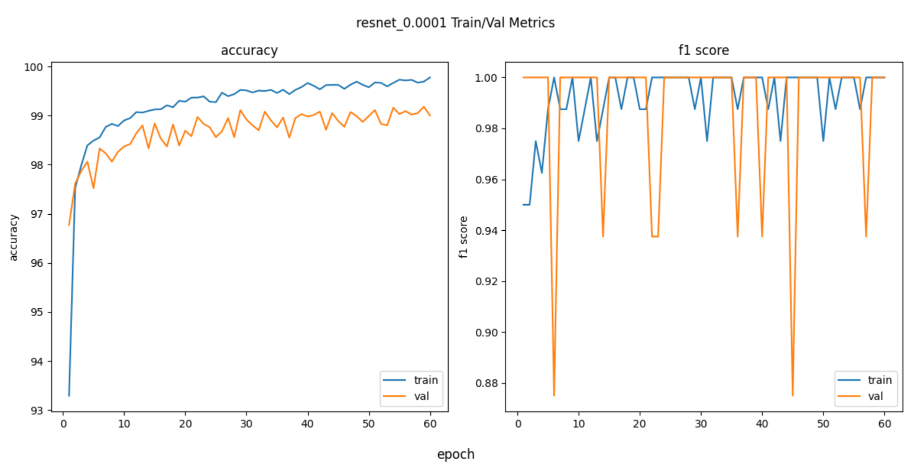

Figure 6: Training and validation accuracy and F1 scores for resnet model with a learning rate of 0.0001

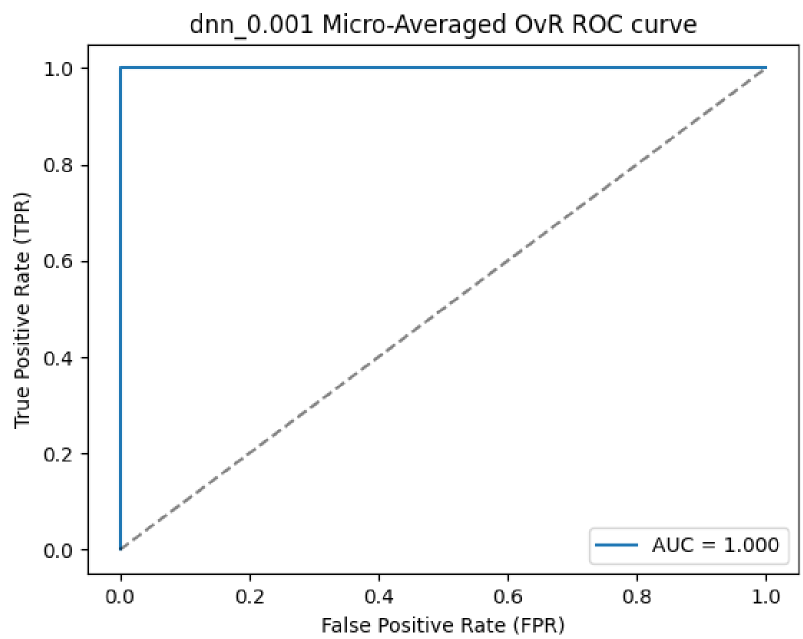

Figure 7: Test ROC curve for dnn model with a learning rate of 0.001

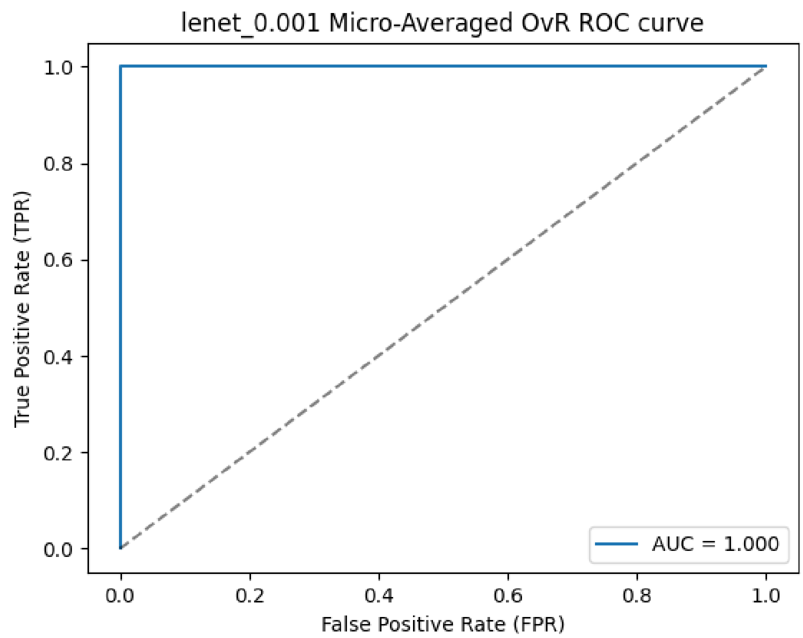

Figure 8: Test ROC curve for lenet model with a learning rate of 0.001

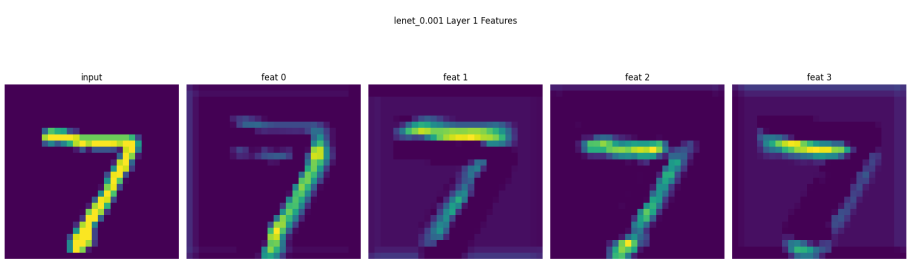

Figure 9: First layer features for lenet model with a learning rate of 0.001

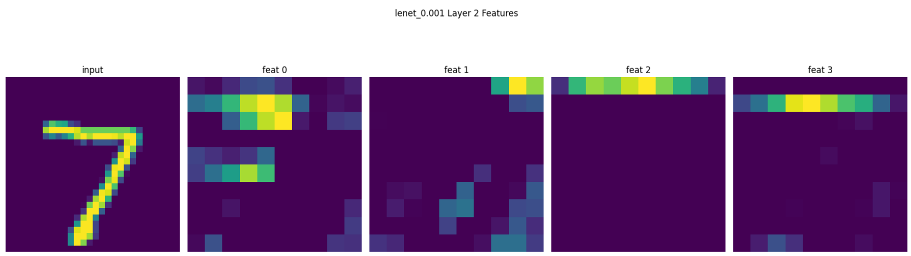

Figure 10: Second layer features for lenet model with a learning rate of 0.001

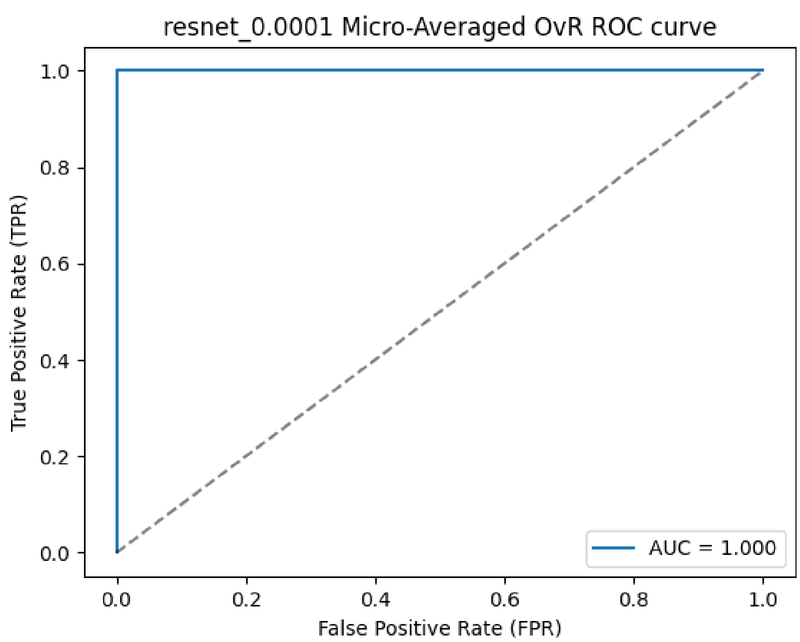

Figure 11: Test ROC curve for resnet model with a learning rate of 0.0001

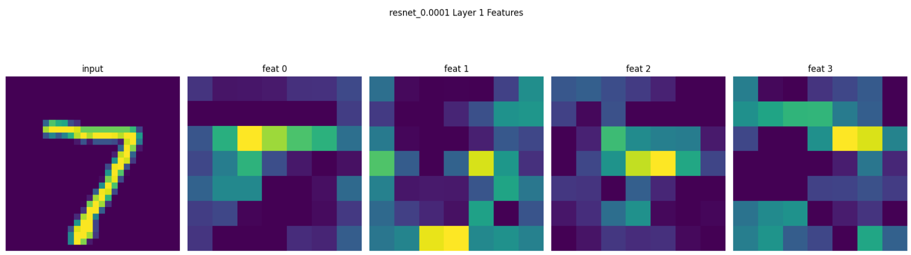

Figure 12: First layer features for resnet model with a learning rate of 0.0001

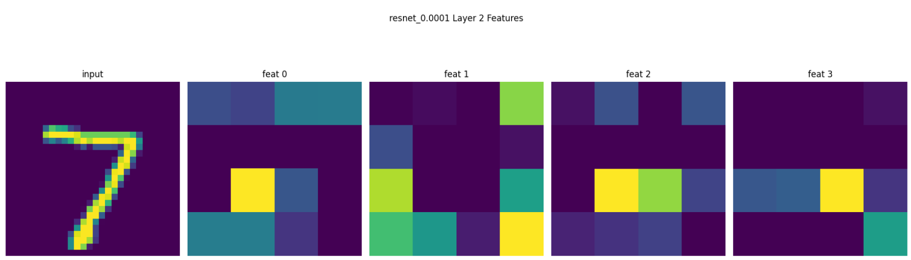

Figure 13: Second layer features for resnet model with a learning rate of 0.0001