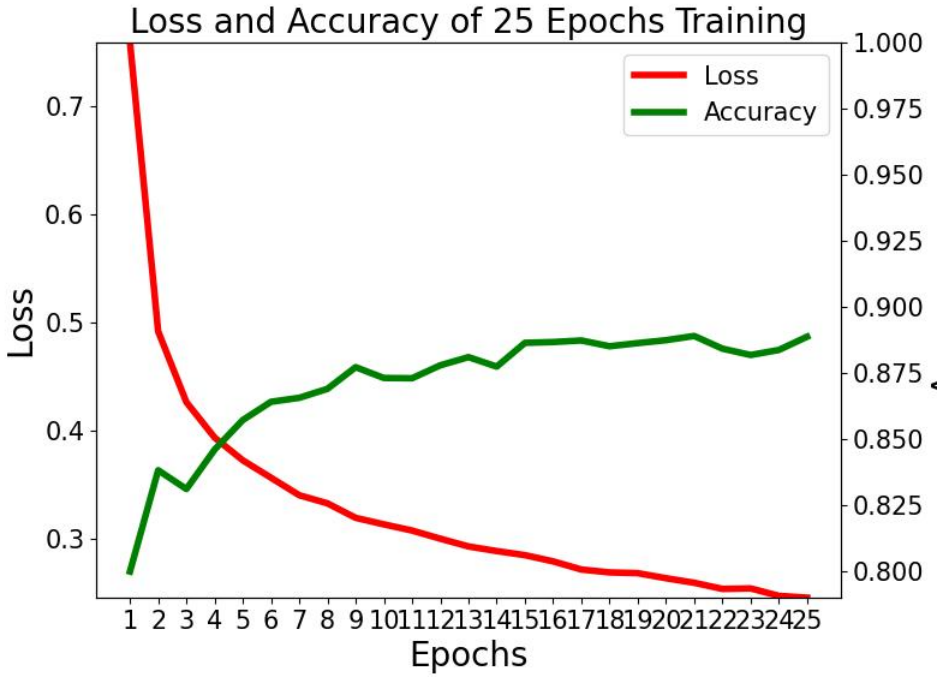
Isaac Stilwell
Professors Zhao & Zheng
Adversarial Machine Learning
1 April 2025

Question 1



As the epoch count increases, the accuracy increases from around 0.800 to around 0.889 and the loss decreases 0.76 to 0.25. Moreover, the initial slopes both have relatively high magnitudes (positive for accuracy and negative for loss), which decrease at higher epochs. Though the overall trend from 0.800 to 0.889 is clear for the accuracy, the graph is fairly volatile and displays a number of local minima and maxima, especially as the trend begins to plateau after epoch 9. In contrast to the accuracy, the trend of the loss is very smooth, with minor bumps occurring at epochs 8, 17, 19, and 23.

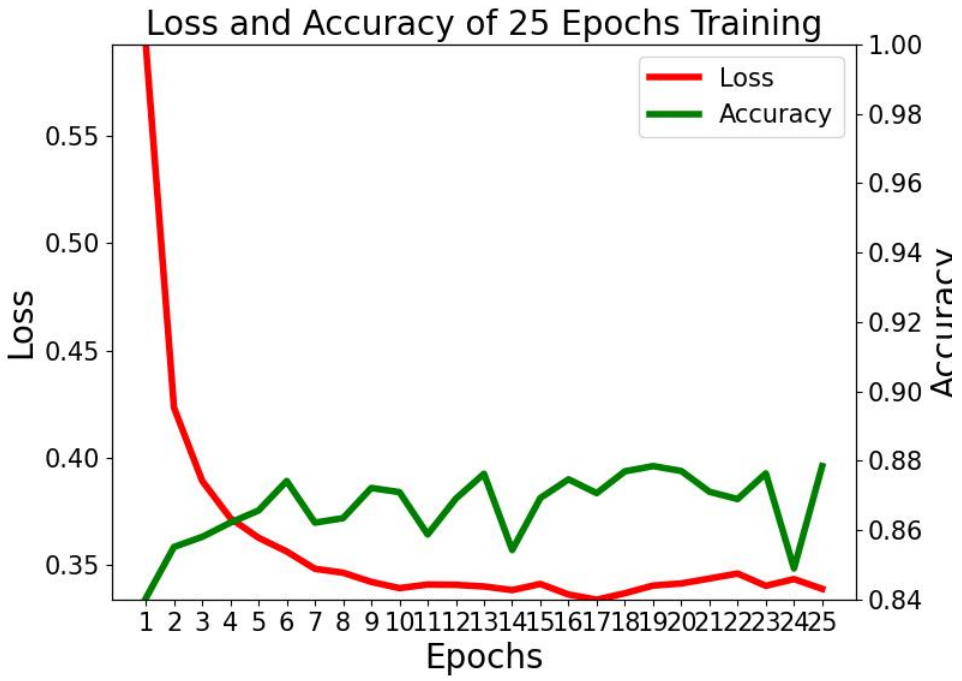
Question 2

```
train_transform = transforms.Compose([
    transforms.ToTensor(),
    transforms.Normalize((0.5,), (0.5,)),
    transforms.RandomHorizontalFlip(),
    transforms.RandomVerticalFlip(),
])
```

The training data is first transformed into a tensor and then normalized to a mean and standard deviation of 0.5, as directed in the instructions. Two additional transformations are also applied—RandomHorizontalFlip and RandomVerticalFlip, both at their default probabilities of

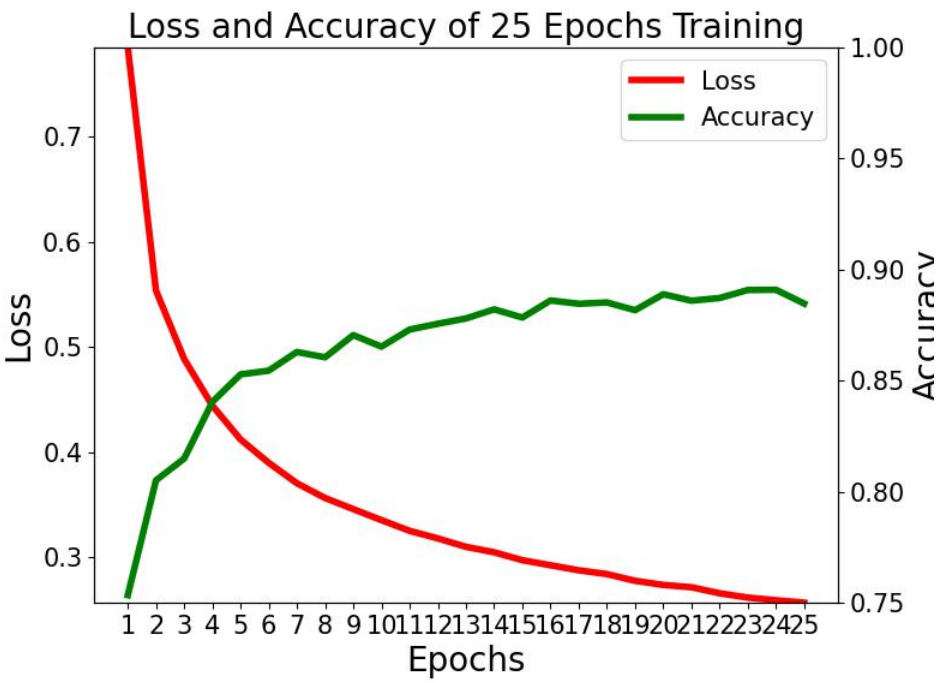
0.5. These augmentations generate additional training data by mirroring existing images in the training set. This helps combat overfitting in the model by reducing sensitivity to orientation of images in the training data and also by simply increasing the size of the training set.

Question 3



To create a comparison model, M2, the batch size was reduced from 64 to 4. M2's performance has similar trends to those of the original model, but with some clear differences. The most obvious is that the starting accuracy of around 0.840 is much higher compared to the original starting value of 0.800. Additionally, the completion accuracy is lower than that of the original model at around 0.879. While the overall accuracy does trend upwards, M2 has much greater variance between epochs, with an especially notable local minimum at the 24th epoch being second lowest accuracy overall. The trend is also less noticeable due to the combination of higher initial accuracy and lower final accuracy. Though the loss shows a similar hyperbolic downward trend, M2's initial and final losses are different at 0.59 and 0.34 compared to the original 0.76 and 0.25. The plateau of the loss is also different, with M2's plateau appearing much flatter than that of the original model.

Question 4



Changing the learning rate from 0.001 to 0.000125 resulted in more similar results to those of the original model, with the exception of the starting accuracy of 0.753 being lower than the original 0.800. Interestingly, a learning rate of 0.000125 performed better than 0.0000625, despite the latter matching the ratio of $\frac{1}{16}$ between the original batch size (64) and the M2/M3 batch sizes (4).

Question 5

The average epoch training speed for the original model was 7.62 seconds. The average epoch training speed for M2 was much higher at 13.26 seconds.

Bonus

To compute the total number of inputs for the first linear layer (which takes the output of the second convolutional layer), the number of channels is multiplied by the number of pixels in the convolved image using eq. (1):

$$number\ of\ feature\ maps \times feature\ map\ size = inputs \quad (1)$$

The instructions provide that there are 16 feature maps in the output of the second convolutional layer. To find the feature map size at this step, the convolution of both layers must be considered.

The first layer starts with a single feature map with the image's dimensions, 28×28. Applying a kernel of size 5×5 (as provided in the instructions) yields 6 24×24 feature maps. Applying

max pooling with a kernel size 2×2 and a stride of 2 to each map yields 6 12×12 feature maps, which is the input for the second layer. Applying the same steps and parameters yields 16 8×8 feature maps after convolution, giving a final output of 16 4×4 feature maps after max pooling.

Finally, substituting the number of feature maps and their corresponding size into eq. (1) returns

$$16 \times 4 \times 4 = inputs$$