Network Architecture

- 2-dimensional convolutional layer with 32 filters and a 5×5 kernel
- 2×2 max pooling layer
- fully connected layer with 128 neurons and ReLU activation
- output layer with 10 neurons and softmax activation

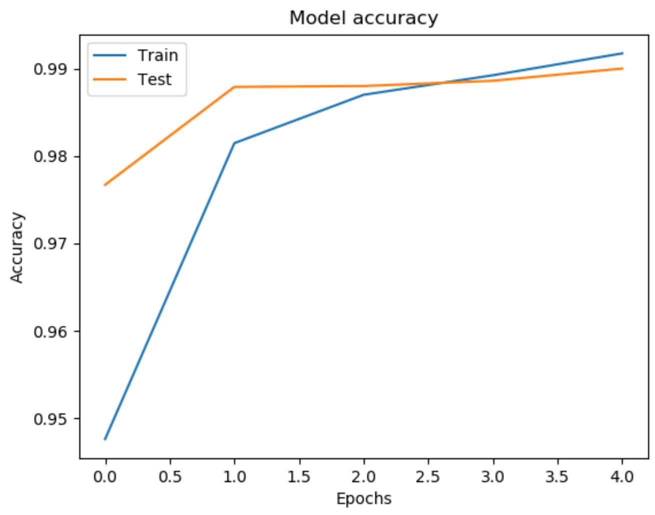

Figure 1: Train and test accuracy over number of epochs.

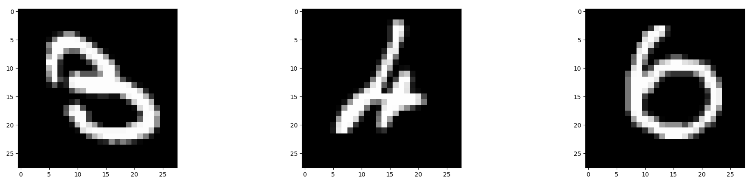

Figure 2: Misclassified examples.

Performance The model is trained for 5 epochs. The train accuracy reached 100% while the test accuracy reached 99% (Figure 1). I was surprised by how quickly the model was able to reach such high accuracy. Magnitude of bias values tend to be smaller than magnitude of values in the weight matrices. The mean for both bias vectors and weight matrices are around 0, but the values themselves are not all 0 (Figure 3).

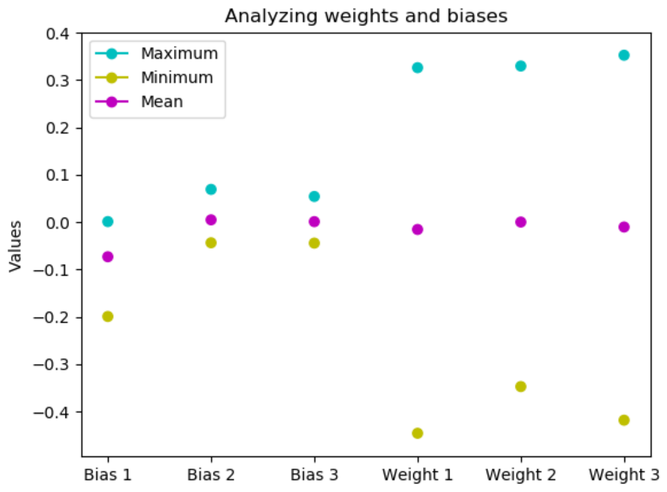

Figure 3: Final weights and biases of trained model. Each weight and bias has 3 points associated with it: maximum, minimum, and mean. These values indicate the largest, smallest, and average of values in the matrix/vector.

Figure 2 shows misclassified examples in the test set. The left image is supposed to be labeled "3", but our model predicts a "5" instead. The center image is labeled "4", but our model predicts a "6" instead. The right image is labeled "6", but our model predicts a "0" instead. This can certainly be confusing to classify since the "6" is basically like a "0" but with a small stem branching out from the top.

Explanation The motivation for using a CNN is that we are classifying an image dataset. It turns out that the CNN works really well for the MNIST dataset. The convolutional layer can extract features from images through convolving with the filter. Then, the pooling layer can reduce dimensionality of the feature map and dropout can prevent overfitting. The ReLU activation function can introduce nonlinearity to the model. Finally, the softmax function can output a probability distribution over the classes. Since the labels are one-hot encoded, then ideally the softmax also outputs the same one-hot encodings. This means that the input has a probability of 1 of being in that correct class.

The loss function is categorical cross-entropy. The advantage of this loss is that the network can learn faster if error is higher. For a large dataset like MNIST, this is important because learning can take very long until the network converges. This network only took about 10 minutes to train on a CPU.