Project 3 Report
By Kushal Gandhi

Model and Training Procedure Description

Experiment 1: Number of Epochs = 100, Batch-Size = 64
    Layer 1: Number of Neurons = 10
        weight initialization scheme = he_normal
        activation function = relu
    Layer 2: Number of Neurons = 10
        weight initialization scheme = he_normal
        activation function = softmax
    Accuracy: 11.19%

Experiment 2: Number of Epochs = 100, Batch-Size = 128
    Layer 1: Number of Neurons = 100
        weight initialization scheme = he_normal
        activation function = relu
    Layer 2: Number of Neurons = 100
        weight initialization scheme = he_normal
        activation function = sigmoid
    Layer 3: Number of Neurons = 10
        weight initialization scheme = he_normal
        activation function = softmax
    Accuracy: 89.18%

Experiment 3: Number of Epochs = 100, Batch-Size = 128
    Layer 1: Number of Neurons = 100
        weight initialization scheme = he_normal
        activation function = relu
    Layer 2: Number of Neurons = 100
        weight initialization scheme = he_normal
        activation function = selu
    Layer 3: Number of Neurons = 10
        weight initialization scheme = he_normal
        activation function = softmax
    Accuracy: 9.2%

Experiment 4: Number of Epochs = 100, Batch-Size = 128
    Layer 1: Number of Neurons = 100
        weight initialization scheme = he_normal
        activation function = relu
    Layer 2: Number of Neurons = 100

weight initialization scheme = he_normal
activation function = tanh
Layer 3: Number of Neurons = 100
weight initialization scheme = he_normal
activation function = tanh
Layer 4: Number of Neurons = 10
weight initialization scheme = he_normal
activation function = softmax
Accuracy: 92.32%

Experiment 5: Number of Epochs = 100, Batch-Size = 128
Layer 1: Number of Neurons = 100
weight initialization scheme = he_normal
activation function = relu
Layer 2: Number of Neurons = 100
weight initialization scheme = he_normal
activation function = tanh
Layer 3: Number of Neurons = 100
weight initialization scheme = he_normal
activation function = tanh
Layer 4: Number of Neurons = 100
weight initialization scheme = he_normal
activation function = relu
Layer 5: Number of Neurons = 10
weight initialization scheme = he_normal
activation function = softmax
Accuracy: 91.40%

Experiment 6: Number of Epochs = 100, Batch-Size = 128
Layer 1: Number of Neurons = 100
weight initialization scheme = he_normal
activation function = selu
Layer 2: Number of Neurons = 100
weight initialization scheme = he_normal
activation function = tanh
Layer 3: Number of Neurons = 100
weight initialization scheme = he_normal
activation function = tanh
Layer 4: Number of Neurons = 100
weight initialization scheme = he_normal
activation function = relu
Layer 5: Number of Neurons = 10
weight initialization scheme = he_normal
activation function = softmax

Accuracy: 91.15%

Experiment 7: Number of Epochs = 200, Batch-Size = 128
   Layer 1: Number of Neurons = 100
      weight initialization scheme = he_normal
      activation function = relu
   Layer 2: Number of Neurons = 100
      weight initialization scheme = he_normal
      activation function = tanh
   Layer 3: Number of Neurons = 100
      weight initialization scheme = he_normal
      activation function = tanh
   Layer 4: Number of Neurons = 100
      weight initialization scheme = he_normal
      activation function = relu
   Layer 5: Number of Neurons = 10
      weight initialization scheme = he_normal
      activation function = softmax
   Accuracy: 92.87%

Experiment 8: Number of Epochs = 500, Batch-Size = 128
   Layer 1: Number of Neurons = 100
      weight initialization scheme = he_normal
      activation function = relu
   Layer 2: Number of Neurons = 100
      weight initialization scheme = he_normal
      activation function = tanh
   Layer 3: Number of Neurons = 100
      weight initialization scheme = he_normal
      activation function = tanh
   Layer 4: Number of Neurons = 100
      weight initialization scheme = he_normal
      activation function = relu
   Layer 5: Number of Neurons = 10
      weight initialization scheme = he_normal
      activation function = softmax
   Accuracy: 91.40%

Experiment 9: Number of Epochs = 500, Batch-Size = 128
   Layer 1: Number of Neurons = 100
      weight initialization scheme = he_normal
      activation function = selu
   Layer 2: Number of Neurons = 100
      weight initialization scheme = he_normal

activation function = tanh
Layer 3: Number of Neurons = 100
weight initialization scheme = he_normal
activation function = tanh
Layer 4: Number of Neurons = 100
weight initialization scheme = he_normal
activation function = relu
Layer 5: Number of Neurons = 10
weight initialization scheme = he_normal
activation function = softmax
Accuracy: 93.55%

Model Performance and Confusion Matrix

Experiment 9 gave the best results so far in terms of accuracy. The architecture of the model is described in the previous section. Find in *Model and Training Procedure Description: Experiment 9*. And following are the performance stats and confusion matrix of best performing model.

Accuracy Score: 0.931161647203442

Precision Score: [0.9573170731707317, 0.9473684210526315, 0.9371069182389937, 0.8895348837209303, 0.9386503067484663, 0.918918918918919, 0.9281437125748503, 0.9418604651162791, 0.8827586206896552, 0.9659863945578231]

Recall Score: [0.9631901840490797, 0.989010989010989, 0.9371069182389937, 0.9107142857142857, 0.9329268292682927, 0.9577464788732394, 0.9337349397590361, 0.9418604651162791, 0.8533333333333334, 0.8819875776397516]

Confusion matrix of your best performing model:

```
[[157  0  0  1  0  2  0  2  1  0]
 [ 0 180  1  1  0  0  0  0  0  0]
 [ 0  1 149  2  1  0  1  1  4  0]
 [ 0  1  4 153  0  4  1  3  1  1]
 [ 0  0  0  0 153  1  6  1  1  2]
 [ 0  0  0  3  0 136  0  0  3  0]
 [ 6  0  1  0  1  2 155  0  1  0]
 [ 0  3  0  2  2  0  0 162  2  1]
 [ 0  3  4  7  0  2  4  1 128  1]
 [ 1  2  0  3  6  1  0  2  4 142]]
```

Training performance plot
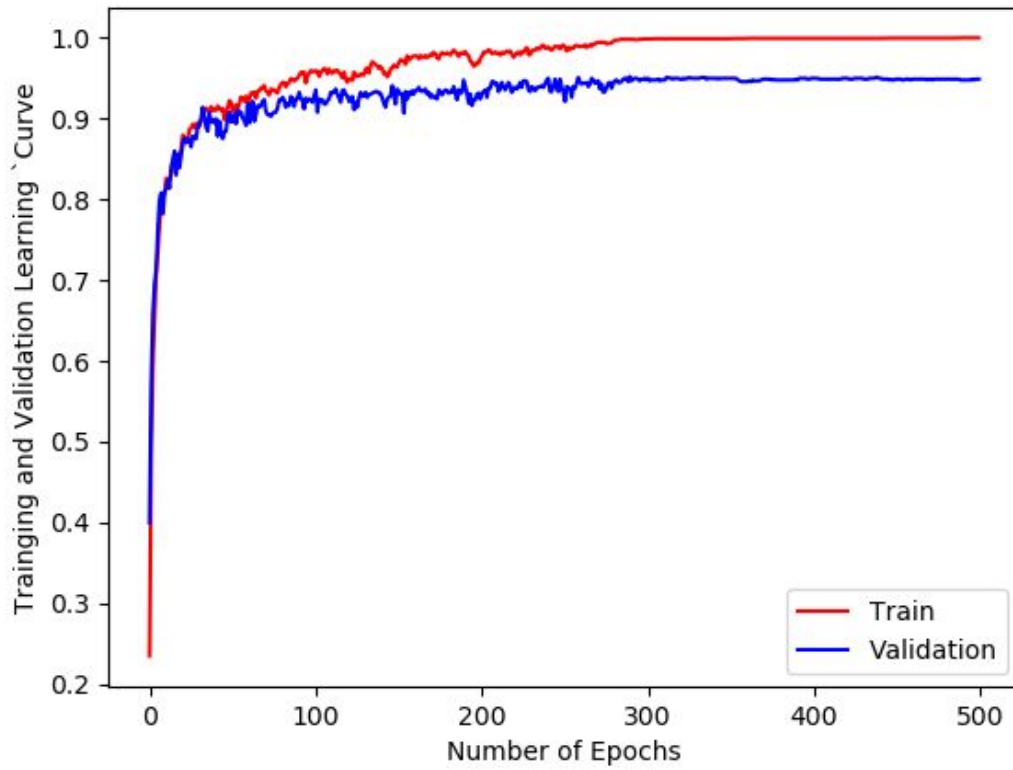
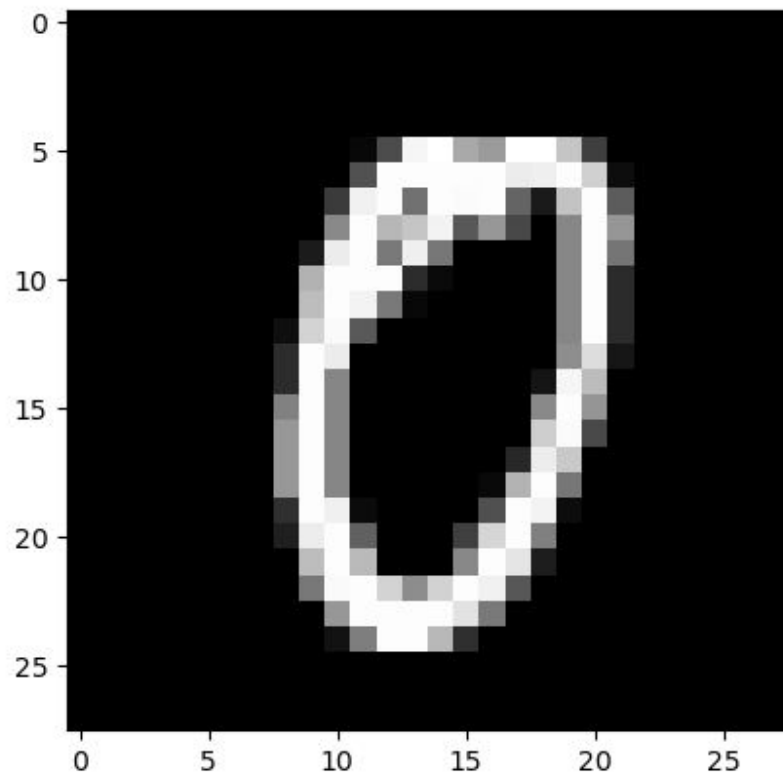**Visualization of three misclassified images:**
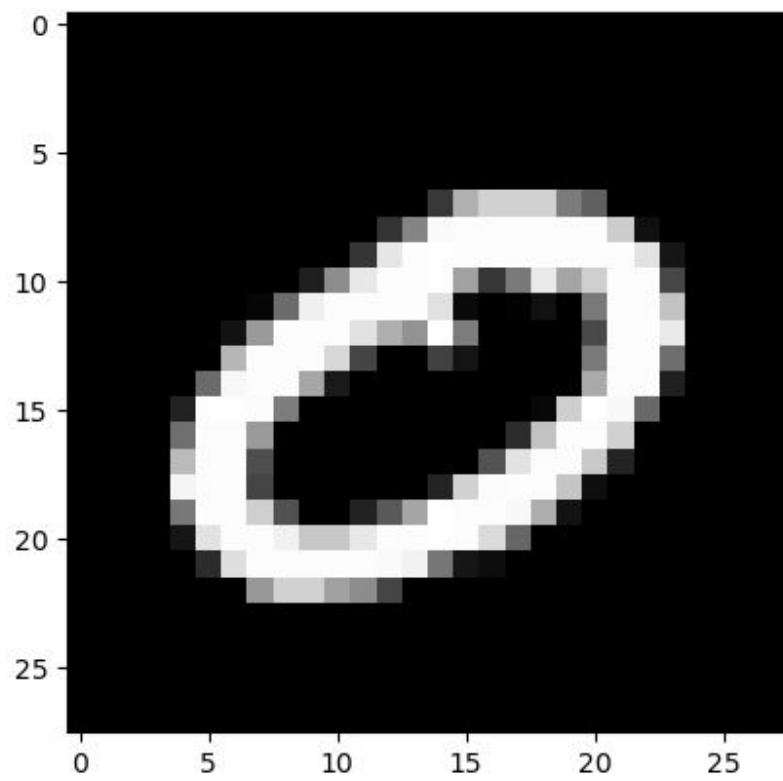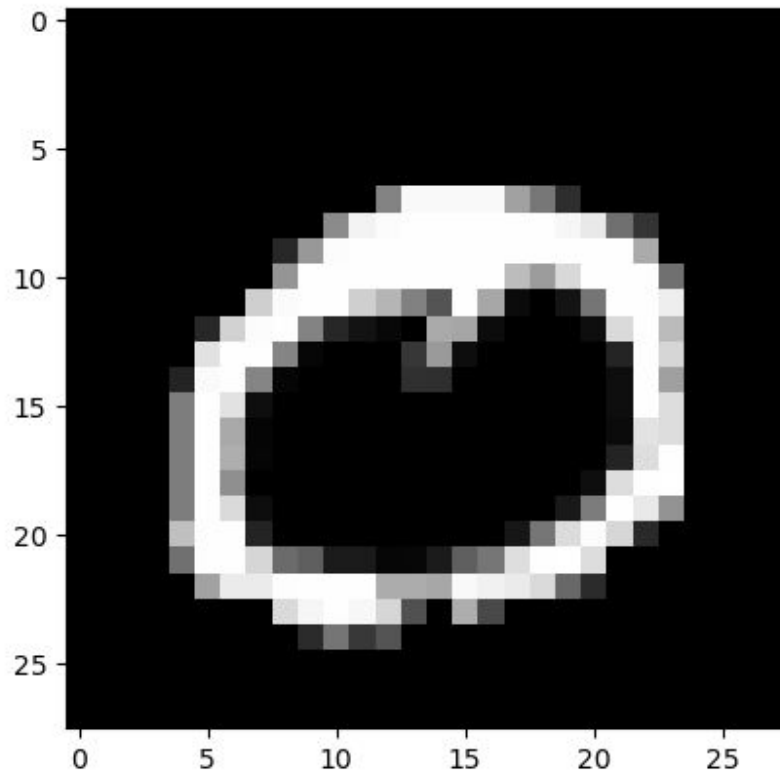
Image 1:

Image 2:

Image 3:

There could be several reasons because of which the model miss-classified these images as some other numbers. One reason could be extra extruding parts that can be seen forming on the inside of the '0', appearing in Image 2 and Image 3. Another reason could be orientation of the number. Sometimes the way and direction in which the numbers are aligned, becomes the cause to accidentally perceive it as a different number.