Kevin Gutierrez – CSCI 495 Deep Learning – Research Assignment

**Introduction**

The purpose of this lab report is to describe the process of creating a neural network in python and training it using the EMNIST letters dataset in order to predict handwritten letter values

**Equipment List**

1. Visual Studio Code

**Overview**

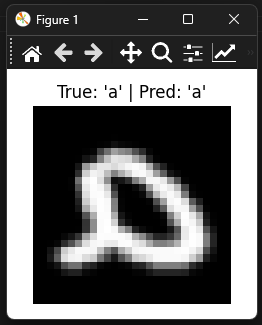
A neural network was requested to be developed and trained on a particular dataset. I elected to have the neural network classify handwritten letters using the EMNIST dataset. This means that there are 26 possible classes, one for each letter of the alphabet.

The EMNIST handwritten alphabet dataset was chosen due to its ease of accessibility and similarity between the MNIST handwritten digit dataset. Thus, the logic covered in the lectures could be used as a basis for developing this neural network. The biggest differences between my neural network and the one covered in the lectures are that my neural network:

1. uses the ReLU over the tanh hidden layer function, to avoid vanishing gradients and help with speeding up learning
2. Uses the softmax instead of the sigmoid activation function in the output layer – this is because the softmax function ensures that the output sum is equal to 1. This helps with picking the most probable letter
3. Uses the categorical crossentropy error function over the mean squared error function as it works better for classification
4. Uses 200 neurons
5. Uses 8 epochs

I implemented functionality to predict the EMNIST test data values, and also my own handwritten letters in ./MyHandwrittenLetters.

**Problems encountered**

At first, while using my own handwritten letters, I couldn’t get the model to predict any of my handwritten letters at all. Then I realized that the model was fed mirrored letters in this format:  
 Notice how the image is mirrored. Also notice how the letter is white and the background is black – that was another source of error in this project. Once I programmatically mirrored my letters and inverted the colors, I was able to get predictions to work.

I also encountered a very low accuracy rate for my own handwritten letters – this was improved by increasing the epochs and number of neurons. I believe this was caused due to my letters being

**Example Run**

In this run, I used EMNIST test data letters

A screenshot of a computer

AI-generated content may be incorrect.

In this run, I used my own personal letters. Here are a few of them:  
A screenshot of a computer

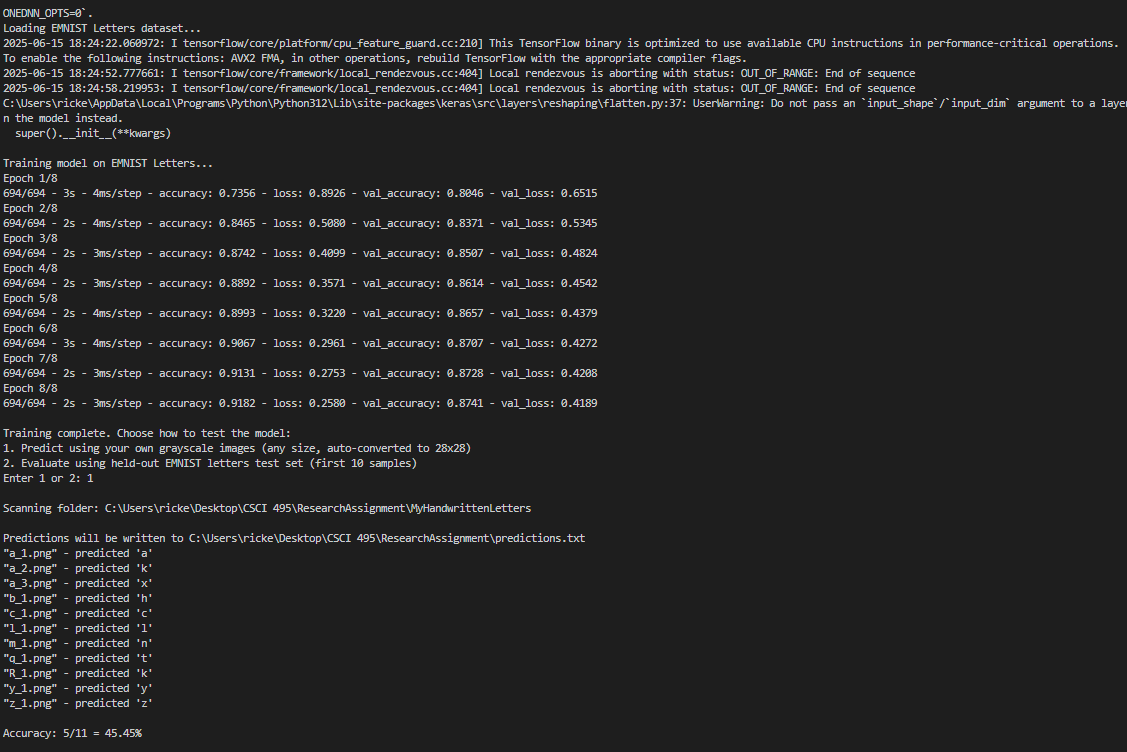
AI-generated content may be incorrect.

A computer screen with white text

AI-generated content may be incorrect.

I got up to 45.45% accuracy with 11 of my handwritten letters:  
A screenshot of a computer program

AI-generated content may be incorrect.



**Conclusion**

This was a phenomenal introduction to developing neural networks. I learned even more about hidden layer functions and their use cases (tahh vs ReLU), and activation functions (softmax vs sigmoid).