Homework 2

Kelsey Hawkins

Introduction

ASL is a prominent language in the American deaf community. With this in mind, I have a dataset of images containing the ASL dictionary, excluding the letters that need motion. The dataset was already flattened into greyscale values, ready for model input with a few preprocessing changes. This model will help translate ASL images to text for those who do not understand it, or want to learn it.

Analysis

The first exploratory item I did was visualizing a few images from the dataset. Not only did I want to ensure the data accuracy, but I wanted to see what resolution I was working with. It was hard to do any summary statistics or heatmaps, etc. due to the data being flattened images so no EDA would have been useful in this case.

However, the images have a 28 x 28 x 1 size, a very low resolution image of hands depicting each letter of the american alphabet. Since the images have been flattened into columns of greyscale values, there will be 784 columns, and all values except for the predictor variable were scaled between 0-1. The predictor variable is a single number that denotes a letter of the alphabet, excluding J and Z which require motion.

```
(27455, 784) (27455, 24)
(7172, 784) (7172, 24)
```

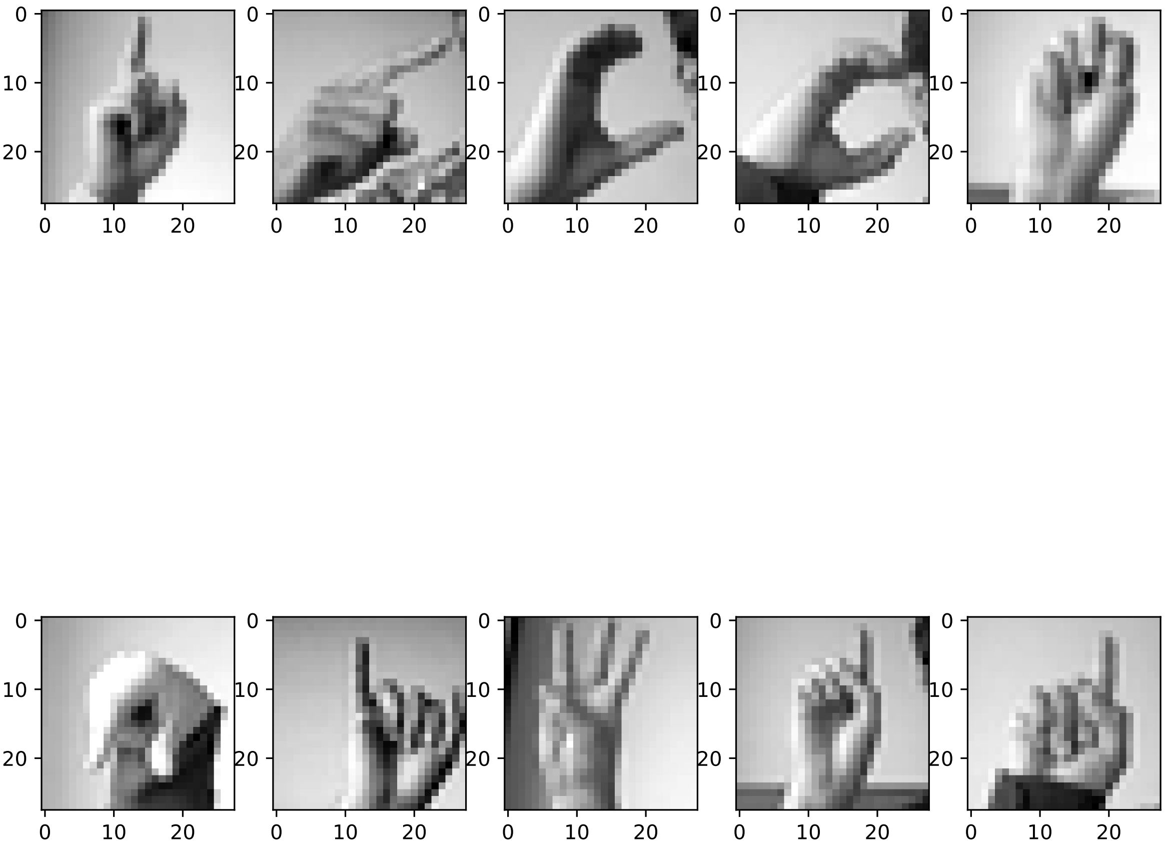

Figure 1: Random Images from Dataset

Methods

Explain the structure of your model and your approach to building it. This can also include changes you made to your model in the process of building it. Someone should be able to read your methods section and generally be able to tell exactly what architechture you used. However REMEMBER that this should be geared towards an audience who might not understand Keras code.

The first model I created is a Deep Feed-Forward Neural Network. I started with a simple model, built in class that had about 6 hidden layers. I then expanded it by adding more dropout, dense, and batch normalization layers. Each dense layer has an activation function of reLU, and a kernel regularizer of L2 on two early dense layers. The last dense layer outputs 24 nodes with a softmax activation function for predicted probabilities for each letter. When I compiled my model, I tested multiple optimizers including SGD, Adam, and RMSprop. The model included early stopping to prevent overfitting.

Results

Detailed discussion of how your model performed. Include a discussion about whether or not Deep Learning was necessary in this situation.

Reflection

In this assignment, I did not come across any struggles or problems with the code or analysis. However I did refer back to the math of the models to justify my answers and think about why the models performed the way they did. It was interesting to be able to quantify them with a simple classification problem and think about how each model handles class imbalances in the slightest way, by about 20%. In the future, I would approach it the same way as I did here, as it seemed to work well.