CSID: Hoque

Link to notebook: http://localhost:8888/notebooks/ Tasneem_Hoque_Assignment4.ipynb

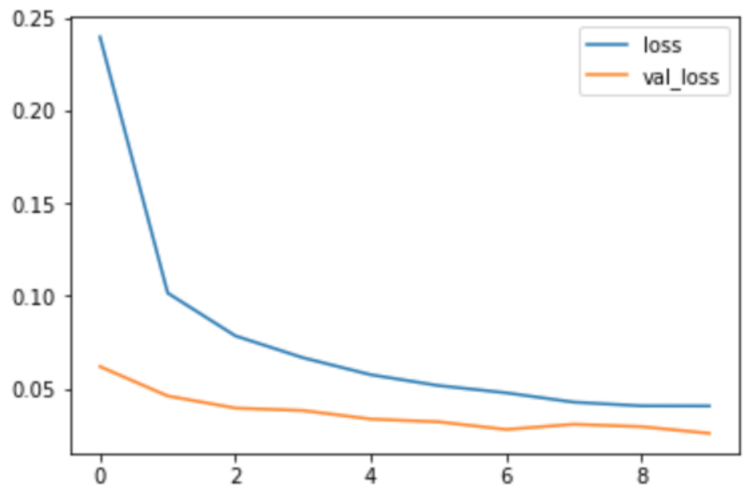
The first step is to install all the necessary libraries to import the data, build the model and test the model. Then I loaded the data into the train and test sets from the mnist data set. I checked the shape of the test and train x sets. They were an 85 to 15% split between test and train sets. Each image is 28x28 pixels in size. The image is resized to 28x28x1, 1 is to account for the color scheme which would be 3 for RGB and 1 for this one, since this image is gray scaled. The x test and train values are then normalized, they are then one-hot encoded.

The image is then reshaped and printed to see a sample with the image label. The image label printed is in one-hot encoded where the 1 is in the 5th position, meaning that the image is of a handprinted 5. In non-one-hot coded format the label is simply the number 5.

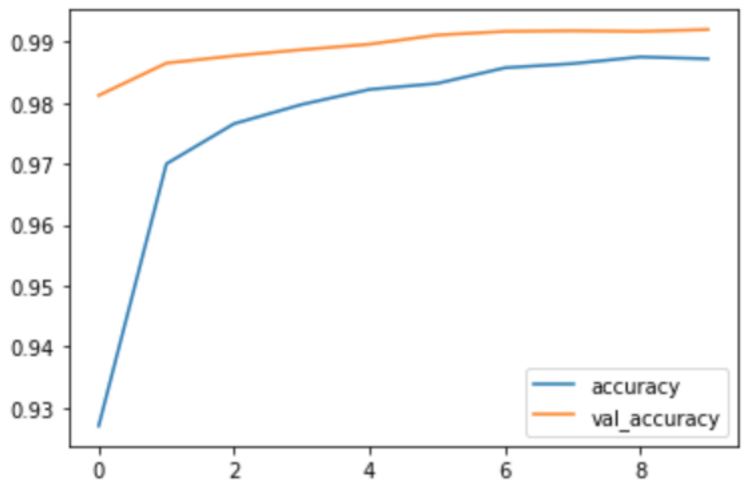
The data is accessed, and we have it formatted, it is time to build the model. The first is a 2D convolution layer with 32 filters of each image, each of those filters are 3x3 in size. A second convolutional layer is added on top of that with 64 filters again with 3x3 windows. Following that, a max pooling layer is added to that takes each 2x2 result to distill the results down to something that is more manageable. A droupout filter is added after to reduce overfitting. The 2D layer is then flattened to a 1D layer and fed into a hidden, flat layer of 128 units. Dropout is applied again to prevent overfitting. Finally, the last layer is made with SoftMax that outputs any one of the 10 categorical answers from 0 to 9.

Once the model is built, it is then time to compile. For this model, I used the Adam optimizer with a relatively small learning rate of 0.001 and for the loss function I used the categorical cross entropy. The model is then fitted with the train and test data with 10 epochs and a small batch size of 32. The metrics for the model is saved within a variable for later evaluation.

A loss and validation loss curve are created, and the results are looking good. The model even passed with a 99.2% accuracy, which is very high.



A loss and validation loss curve are created, and the results are looking good. From the loss curve we can see that immediately as the model is run the loss is high, however it gradually decreases by the end of the last epoch, since the model is better trained. Loss for validation is constant, but slightly decreasing as seen from the graph.



An accuracy and validation accuracy curve are created, and the results are looking good. The model even passed with a 99.2% accuracy, which is very high. From the accuracy curve we can see that immediately as the model is run the accuracy is low, however it gradually increases by the end of the last epoch, since the model is better trained. Accuracy for validation is constant, but slightly increasing as seen from the graph.

The last thing to do is to test the model on a single prediction. The model was tested on the second test element, and the number guessed was 1 and it matched with the labeled data. Therefore, the model is successful.