Report

Mid-submission

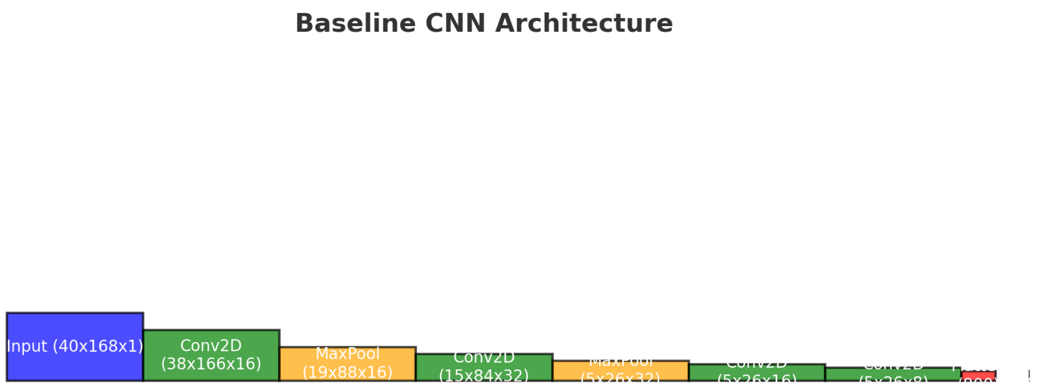
Overview:

Trained a baseline CNN on 50% on the data, kept 30% for test, and 10% for validation, got a validation accuracy of around 9-11% on the dataset, highest validation accuracy that was converged upon was 12%.

Data processing

For each image, I mean-normalized it(so divided by 255, and subtracted the mean from it), and used that image tensor instead. For this problem, i framed it as a classification task, so essentially it became a 37 class prediction task, where given an image, you predict the sum of the digits within that image. As opposed to directly predict the number, i one-hot encoded the sums, and aimed to obtain it's one-hot encoded tensor instead. I made dataloaders for this baseline task, where you essentially obtain the mean-normalized image, and the one-hot encoded sum.

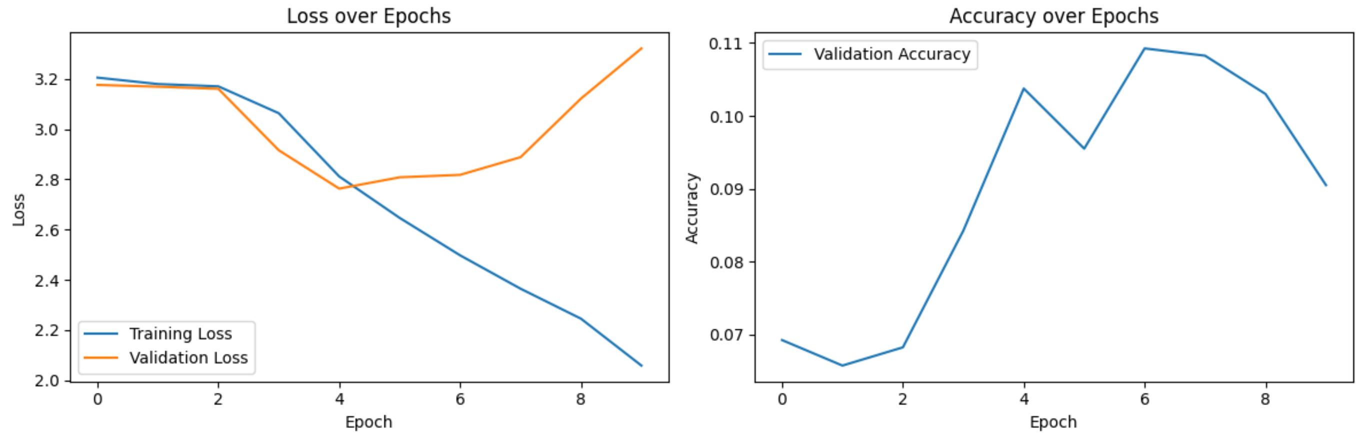
Architecture :

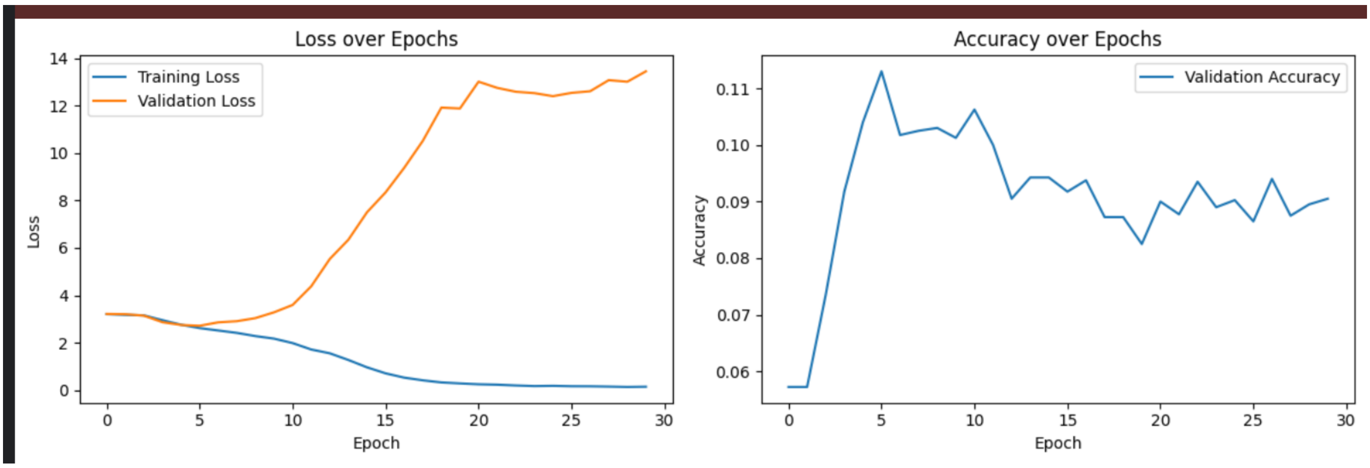


Used a simple set of convolutions followed by a fully connected linear layer, and a softmax, in order to predict the one-hot encoded tensor of the sum of the digits within the image.

Training

For the Baseline CNN, I used cross-entropy loss to optimize over. Usually what I had seen in most of my experiments was, you'd have a maximum validation accuracy at around 12% after which, the model started overfitting, this could've been avoided via early-stopping or drop-out, however since we had to use a vanilla-CNN, decided not to use these.





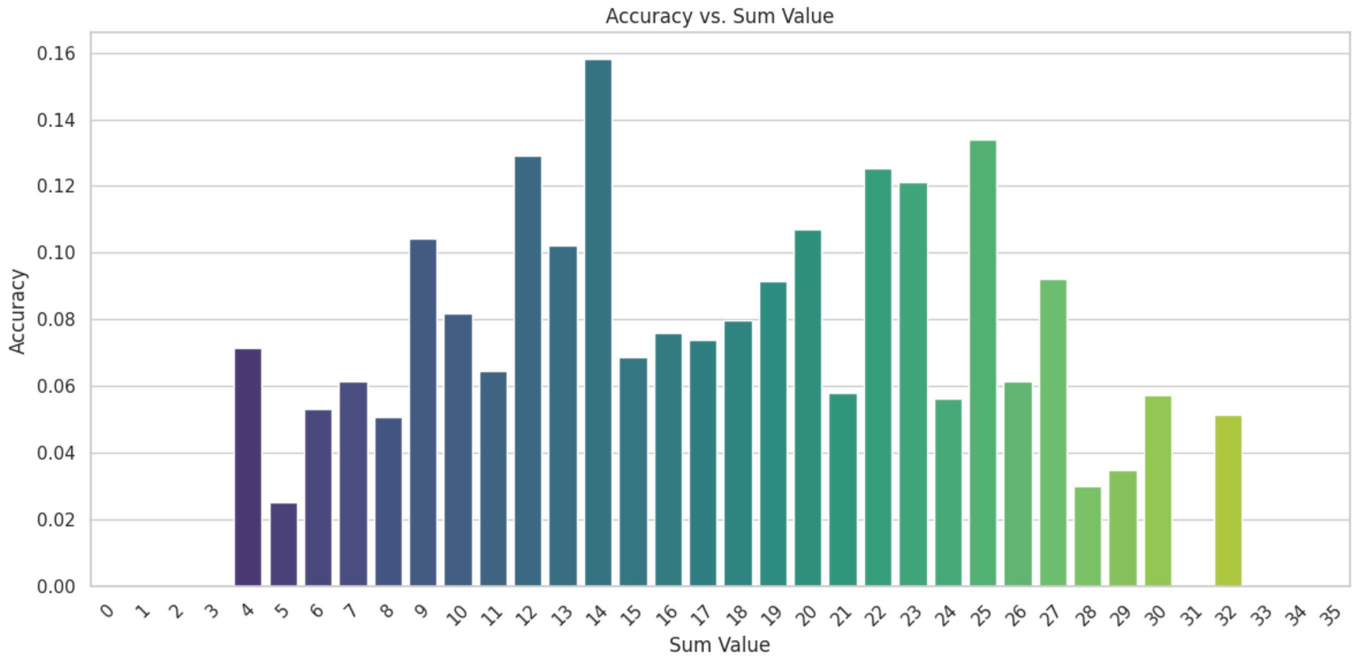
So the validation accuracy for this baseline cnn usually peaked around 11%. The training loss if left for 30 epochs or so, decreased to about 0.8.

Inference

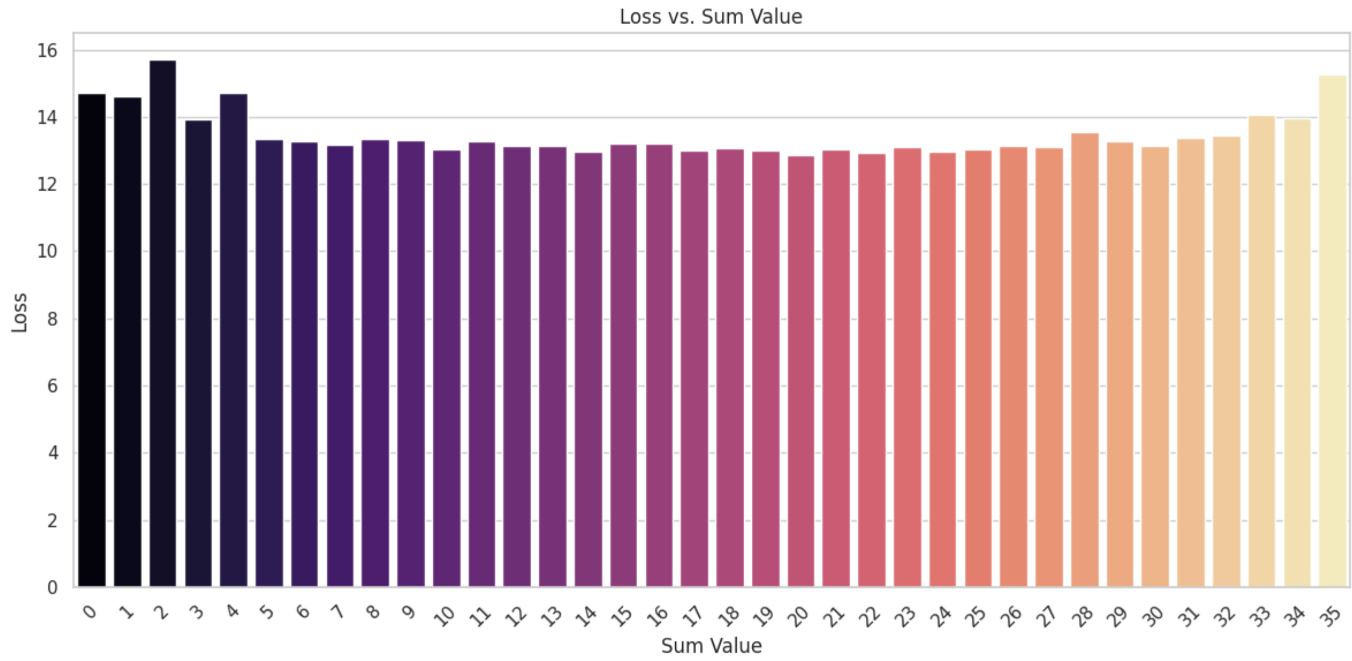
The test results show an average test accuracy of 8.83% and an average test loss approximately in the range of 9-10%. The overall error statistics indicate a mean error of -0.32, a median error of 0.00, and an error range spanning from -17 to 19, with a standard deviation of 4.47. These values suggest that the model's predictions exhibit a wide range of variability.

This suggests the model is unable and struggling to make good predictions, which is expected from a small baseline cnn model, on such a task.

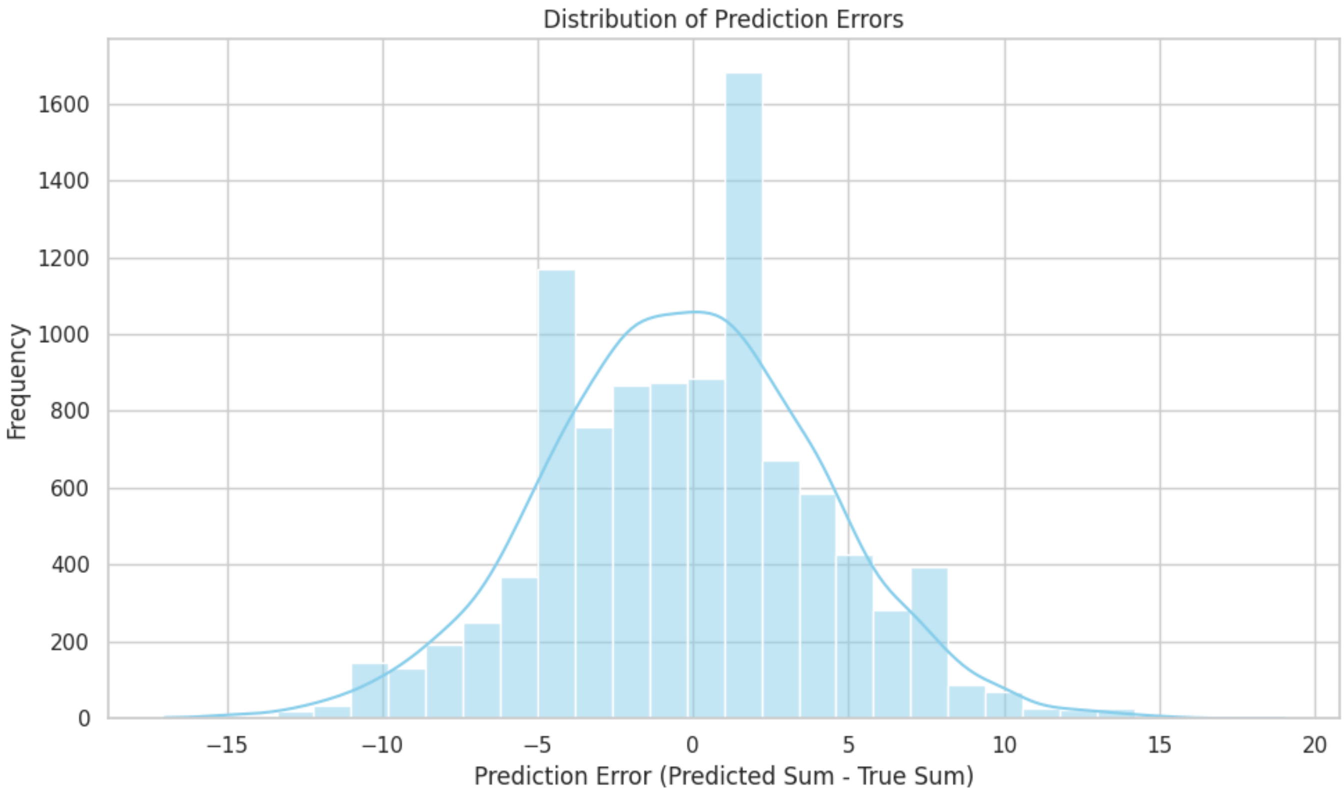
Here are some plots to infer how the model has learned.



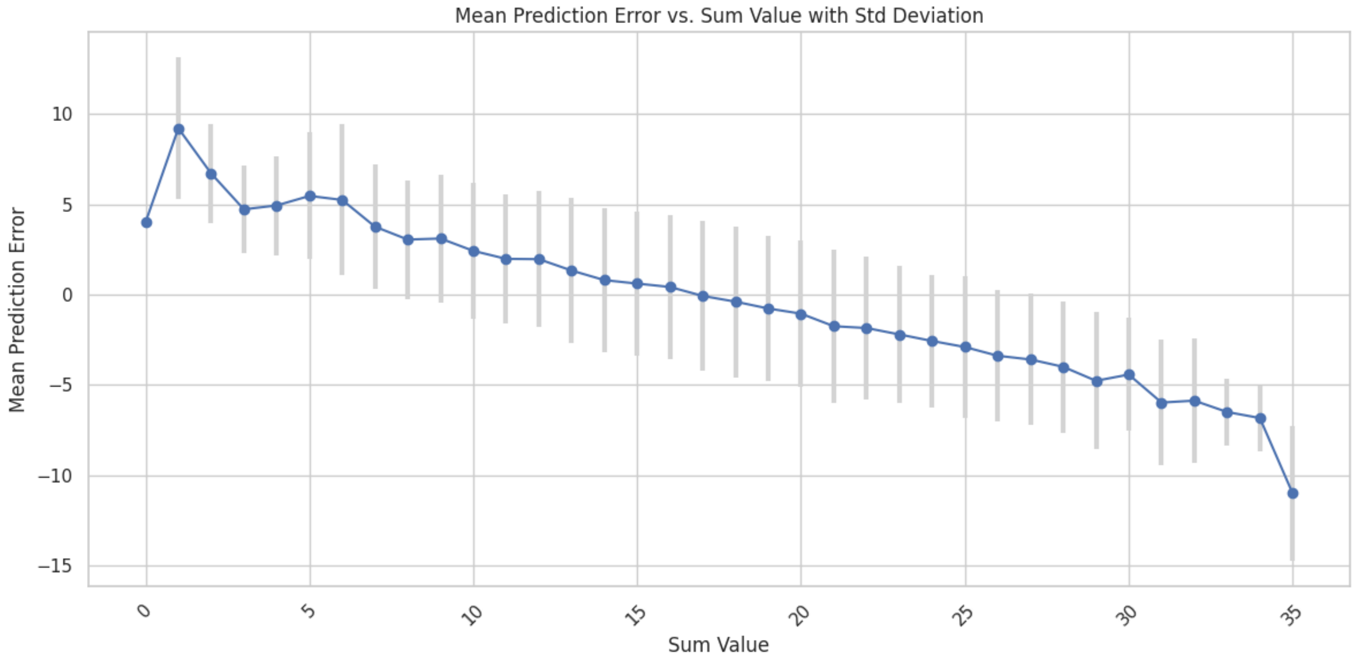
The model has a high accuracy at predicting values within the lower middle part of our sum range, this may be attributed to the fact more values lie within this range for our dataset.



The above plot, is the average loss value, vs the sums present in the dataset, this shows the same observation as in the above diagram, we have a lower loss for the sum value in the middle range, while higher values for values at both ends, which could be because of the distribution of sums within the dataset.



This plot is for the predicted error that our model has vs frequency, from this, we can see it is approximately normal, and we know that around 66% of the time, our error range for the baseline cnn's predictions, will lie in the 4.42 to -4.42 range.



This final plot is the mean prediction error vs the sum value of the dataset. Again it shows the same idea as the above plots, the baseline cnn is able to predict the middle sum range better, simply because more labels in the dataset correspond to within this range.