

MLNS Assignment Part 2

Shreyas Kumar Sinha (2021113010, shrssnh on github)

The model

For part 2, I initially started working with a CNN, same as the previous one but as a classifier. The results were not very promising. I tried various approaches but none were very successful. The rough work can be found in "training_CNN.ipynb"

So I tried using some preprocessing to separate out the digits. This turned out to be extremely challenging, as several of the digits were overlapping. Many different techniques were tried (erosion, dilation, making contours, density based clustering, histogram based splitting, etc) but the overlapping digits made it really difficult, and I ended up giving up in the end. Some residual results are present in "training_filters.ipynb".

Finally, I resorted to using ResNet 18 trained on this dataset. The model and its weights were imported from torchvision's library. The First layer was accommodated to accept 1 channel instead of 3 (our images are grayscale not RGB) and the final fully connected layer was modified to give 41 outputs (I was supposed to go till 37 ($4 \times 9 + 1$) made a silly error, but this does not affect model performance). Just the features extracted from ResNet were sufficient to provide an excellent accuracy of >95%.

Training

The model is trained using PyTorch's Adam optimizer with a learning rate of 0.001 for 40 epochs. Cross Entropy Loss function is used as a measure of the loss. L1 loss is the Mean Absolute Error.

Results

Final ResNet based classification gives us an accuracy > 95% which is very good, and we get a Mean Absolute Error of **0.0299**, which is much better than the baseline CNN which gave a Mean Absolute Error of **1.2624** on the full dataset.