Sahil Rangwala
MSDS422

Week 7 Write-up

This week we explored the MNIST dataset further by conducting more EDA as well as creating different iterations of neural network models ranging from a simple to a more dense neural model. Additionally, graphing and other accuracy metrics were recorded to help compare and contrast the differing models. These metrics were nicely contained in various python functions in the code.

Some EDA insights was that there existed no NULL values or NAN. This means no data cleaning needed to be done to replace the empty values. Additionally, the data was fairly uniform. This means that each number was a label, and the quantity of each label was equally distributed. Additionally, each of the characters are written pretty drastically differently. LIke some characters of 5 are interesting on how people might have written it as each character has a unique interpretation.

The first neural network consisted of 2 layers, and the input layer is 20 nodes and the output layer is 10 nodes. The output layer being 10 nodes makes sense, as there are 10 digits categorical buckets that a character can fall under. This neural network is going to perform the worst, as it does not have any other parameters or further tuning done. However, the basic neural network still had an accuracy of 92%. This performed better than the simpler modeling techniques done in the previous weeks as well as indicate the simplicity and robustness of neural networks.

The next neural network consisted of 3 layers, where an additional wider layer was imputed between the input and output layer. This would help categorize subtle characters into more quantifiable buckets. A worry of mine was that shallow networks are good at memorization but not good at generalization. This would mean that an extremely wide network would be great at recollecting the exact image and what bucket it falls into for the dataset we trained it upon, but on newer data it would struggle. However, the accuracy still fared well with 93% for the semi dense neural network.

The dense neural network consisted of 5 layers and ranging from a funnel of 200, 100, 50, 25, 10 nodes per the respective layer. I believe this approach would fare the best, as it is the perfect balance of number of nodes and number of layers. The advantage of multiple layers is that they can learn features at various levels of abstraction. Multiple layers are much better at generalizing because they learn all the intermediate features between the raw data and the high-level classification. Deep networks already can be very computationally expensive to train, so there's a strong incentive to make them wide enough that they work well, but no wider. In this case, the perfect blend of # of layers and # of nodes led to an accuracy value of 96%.

Each of these accuracy metrics can be visualized from the accuracy/loss graphs, as well as the confusion matrix. The diagonal of the confusion matrix indicates the correctly predicted characters, and shows the number of correctly predicted y values. With these visualizations and the accuracy scores, clearly we can see the predicted 5 layer dense neural network performed the best.

dense_neural_network.csv a minute ago by Sahil Rangwala add submission details	0.95814
semi_dense_neural_network.csv a minute ago by Sahil Rangwala add submission details	0.92835
naive_neural_network.csv 2 minutes ago by Sahil Rangwala add submission details	0.91635