Advanced Machine Learning - Assignment 3

Pranav Kasela
846965

Introduction

In this assignment the objective was to use the CNN with a limited number of parameters (7500) to solve the MNIST number classification problem. Using the ANNs even without using any hidden layers the numbers of parameters jumps to $28 * 28 * 10 + 10 = 7850$, and it yields a maximum accuracy of 88.74%. Using the CNNs with convolution and pooling layers we can reduce the number of parameters and possibly achieving a better result than the ANNs. The data is already divided into training and test set, the training set is splitted into training and validation (80%-20%) to check the model performance during the training, later they will be merged into a single dataset to train the final model with the best hyper parameters.

Model

The model is as follow:

1. Input Layer
2. Convolution - 16 filters, kernel size 3 $\rightarrow 16 * 3^2 + 16 = 160$ parameters
3. Batch Normalization
4. Max Pooling - pool size = 2
5. Convolution - 16 filters, kernel size 3 $\rightarrow 16^2 * 3^2 + 16 = 2320$ parameters
6. Batch Normalization
7. Convolution - 4 filters, kernel size 3 $\rightarrow 4 * 16 * 3^2 + 4 = 580$ parameters
8. Batch Normalization
9. Max Pooling - pool size = 2
10. Flatten - Creates 64 neurons
11. Dense - 58 neurons $\rightarrow 64 * 58 + 58 = 3770$ parameters
12. Dropout of 20%
13. Output - 10 neurons $\rightarrow 58 * 10 + 10 = 590$ parameters

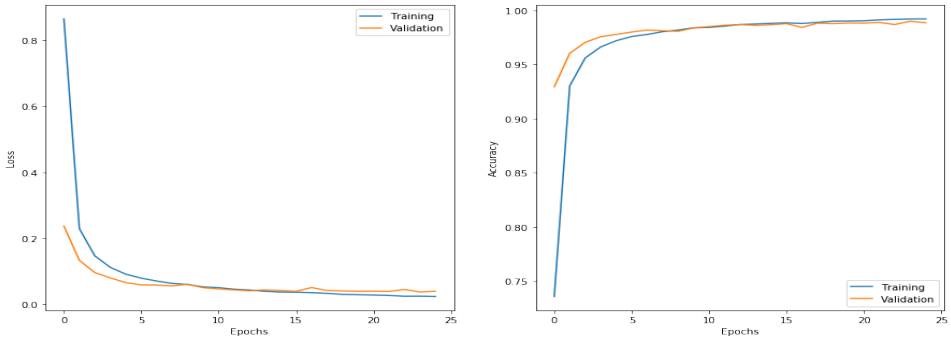

Figure 1: Train and Validation comparison

The number of total parameters are $160 + 2320 + 580 + 3770 + 590 = 7420$. The loss function used is the categorical crossentropy and the optimizer used is Adam with a batch size of 256 and 30 epochs. All the layers are activated with a relu function except for the output layer which has an softmax activation, the learning rate is the default one: 0.001. The model achieves an accuracy of 99.60% in training and 98.85% in validation.

Conclusions

The performance of the test set obtains the result in Figure 2.

	precision	recall	f1-score	support
0	0.9949	0.9939	0.9944	980
1	0.9860	0.9956	0.9908	1135
2	0.9941	0.9855	0.9898	1032
3	0.9921	0.9891	0.9906	1010
4	0.9878	0.9908	0.9893	982
5	0.9855	0.9922	0.9888	892
6	0.9937	0.9885	0.9911	958
7	0.9874	0.9874	0.9874	1028
8	0.9897	0.9887	0.9892	974
9	0.9821	0.9812	0.9817	1009
accuracy			0.9893	10000
macro avg	0.9893	0.9893	0.9893	10000
weighted avg	0.9893	0.9893	0.9893	10000

Figure 2: Test Set classification report

The model achieves an accuracy of almost 98.93%, which is way better than the simple ANNs with a similar number of parameters. Moreover the Figure 1 indicates the model doesn't over or under-fits.