Assignment 1 - Classification Using Convolutional Neural Networks

Aditya Jain

1 Run and Extend to Multi-Class Classification

Q1: The training is run for 35 epochs for the binary classification task and figure 1 shows the plot for training loss.

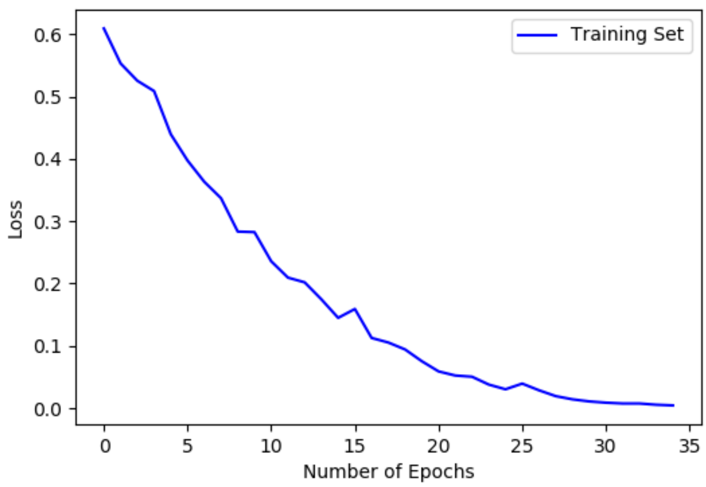

Figure 1: Training loss for the binary classification task

Q2: The classification accuracy on the test set is **80%**

Q3: The training is run for 50 epochs for the multi-class classification task and figure 2 shows the plot for training loss. The test classification accuracy is **60%**.

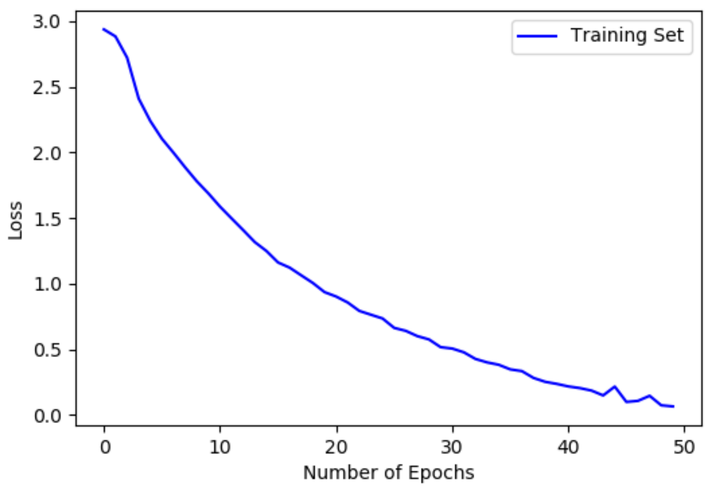

Figure 2: Training loss for the multi-class classification task

2 Change CNN Architecture

Q1: The training is run for 50 epochs for the multi-class classification task with batch normalization. Figure 3 shows the plot for training loss. The test classification accuracy is **67%**.

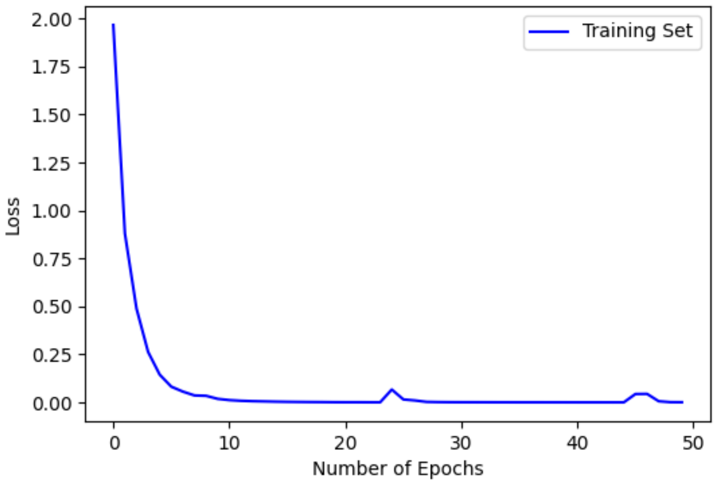

Figure 3: Training loss for the binary classification task with batch normalization

Q2: The training is run for 50 epochs for the multi-class classification task with batch normalization and dropout. Figure 4 is the plot for training loss and the test classification accuracy is **66%**. The accuracy is low because we are over-fitting the training data. The use of validation set should solve the issue.

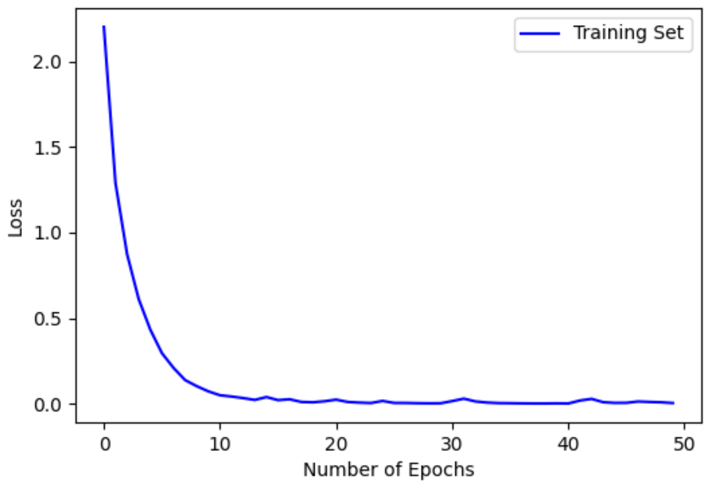

Figure 4: Training loss for the binary classification task with batch normalization and dropout

3 Training Neural Network with Validation

Q1: Figure 5 shows the plot for training and validation loss. The model corresponding to the least validation loss is finally saved and used for evaluation on the test set.

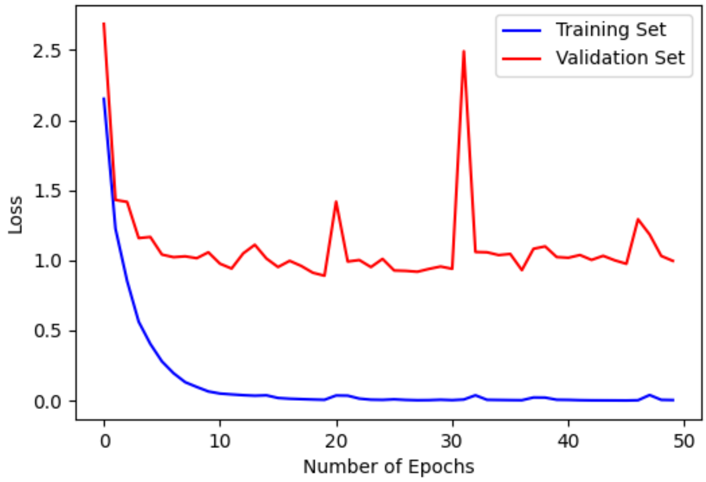

Figure 5: Use of validation loss while training

Q2: The test accuracy is **78%** in this case, compared to the **66-67%** accuracy in the previous parts where the validation data was not used.

4 Hyperparameter Tuning

Q1: Two other optimizers are tested: **Adagrad** and **SGD**. The test classification accuracy for **Adagrad** is **53%** and for **SGD** is **46%**. The training had to be run for 60 epochs because a learning rate of 1e-4 is too slow for Adagrad and SGD. Even after running for 60 epochs, the validation loss didn't converge.

Q2: For **RMSProp** optimizer and a batch size of **32**, the model is tuned for the following learning rates: **1e-2**, **1e-3**, **1e-4**, and **1e-5**. It can be seen in figure 6 that the higher learning rates (1e-2, 1e-3) converge much faster than lower learning rates (1e-4, 1e-5) during early stages of the training. From the graph it seems like the training loss has not converged for 1e-5 and requires further training. There is also high variability in validation loss for 1e-2 because a high learning rate fails to converge and oscillates around the global (or local) optima.

Table 1 shows the test accuracy for the different learning rates. 1e-2 is too high a learning rate to converge whereas 1e-4, 1e-5 are too low to converge, thus requiring further training. Or maybe it's stuck in a local optima. The best accuracy is achieved for 1e-3.

Learning Rate	Test Accuracy
1e-2	58%
1e-3	**71%**
1e-4	65%
1e-5	63%

Table 1: Test accuracy for different learning rates

Q3: For **RMSProp** optimizer and a learning rate of **1e-4**, the model is tuned for the following batch sizes: **16**, **32**, **64**, and **128**. Figure 7 shows the training and validation loss plots. During

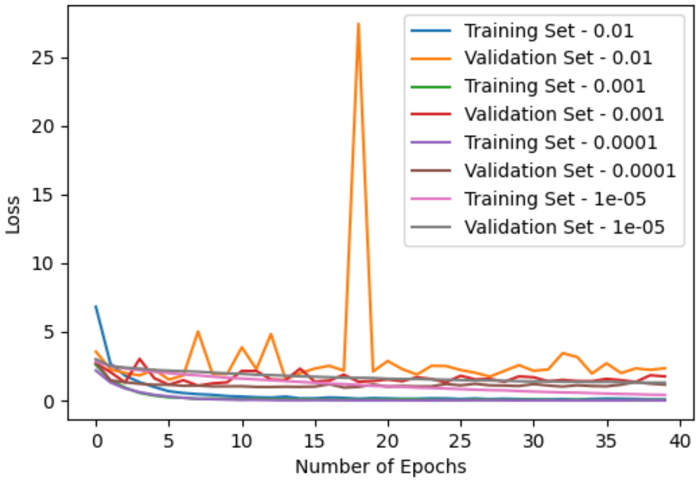

Figure 6: Learning rate tuning

training, it is expected for the smaller batch sizes to converge faster than bigger ones. During evaluation on the validation set, smaller batch sizes (16, 32) have higher variance but low bias and bigger batch size (128) has a higher bias but a lower variance.

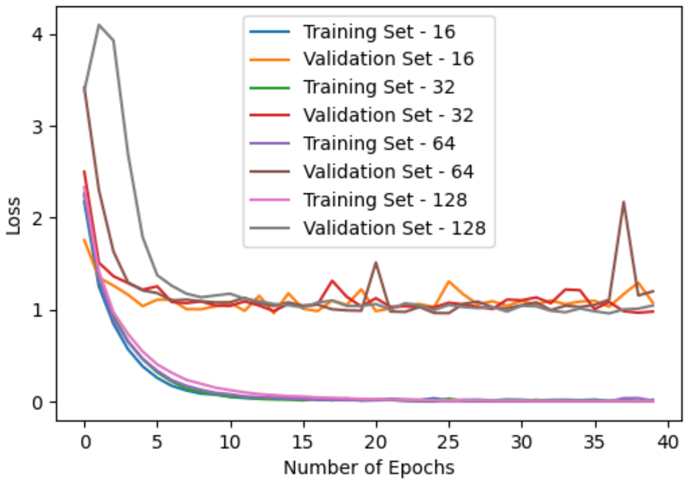

Figure 7: Batch size tuning

Table 2 shows the test accuracy for the different batch sizes and batch size of 64 gives the best result of 76%.

Batch size	Test Accuracy
16	71%
32	67%
64	**76%**
128	66%

Table 2: Test accuracy for different batch sizes

Q4 ***(Extra Credit)***: The models are trained for the following combination of parameters:

- **Optimizer**: RMSProp, Adagrad, and SGD
- **Learning rate**: 1e-2, 1e-3, 1e-4, and 1e-5
- **Batch size**: 16, 32, 64, and 128

This leads to a total of 48 combinations. The best accuracy is achieved for RMSProp, 1e-4, and 64 giving **79%**.