Aniket Sanjay Gujarathi
1007629979

# Assignment 1



## Run and Extend to Multi-Class Classification

### Q1

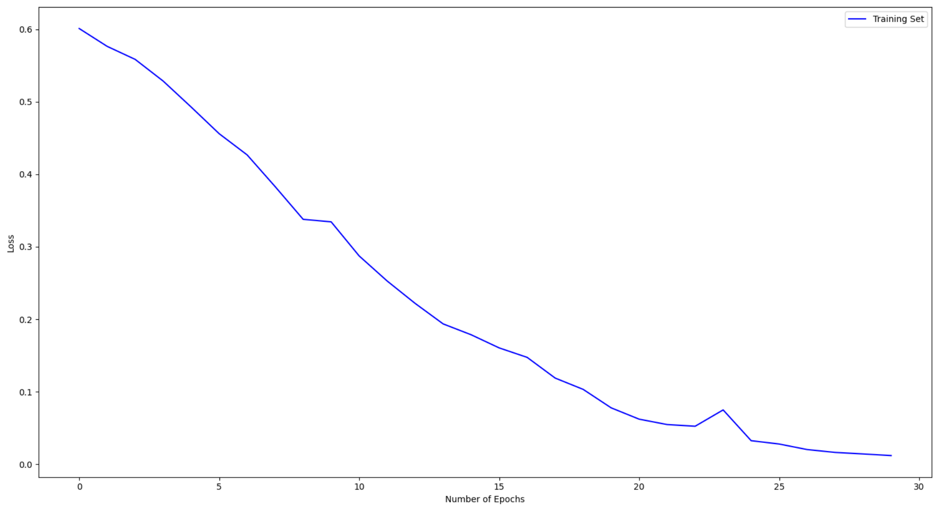


Figure 1: Training Loss for 30 epochs

### Q2

The overall classification accuracy for the vanilla network is 80%

**Q3**

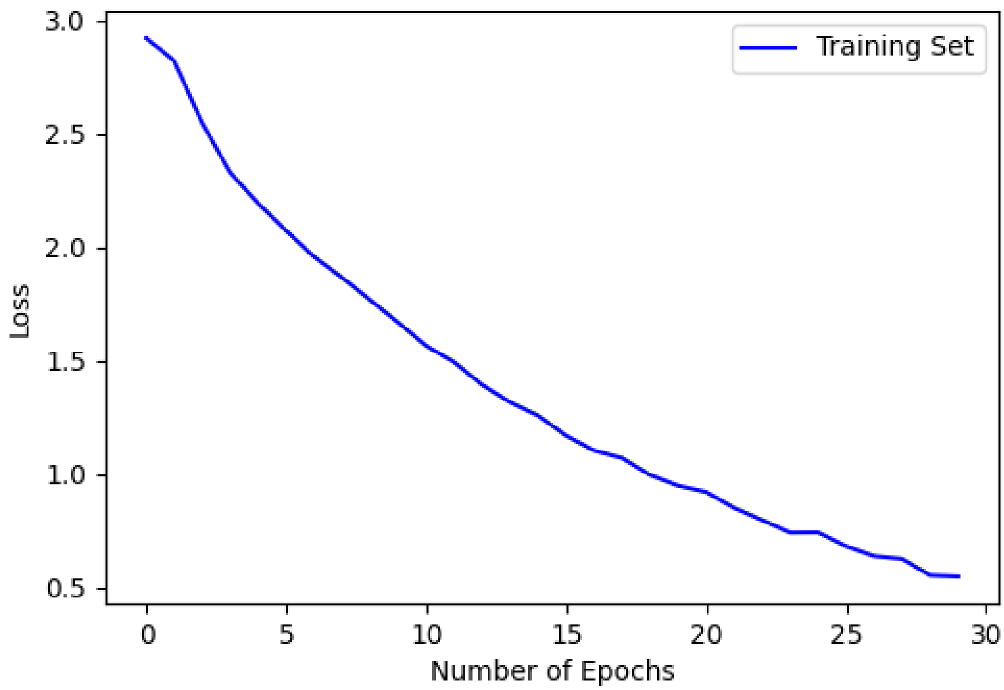


Figure 2: Training Loss for 30 epochs with 20 classes

The overall classification accuracy for 20 classes is 64%

## Change CNN Architecture

### Q1

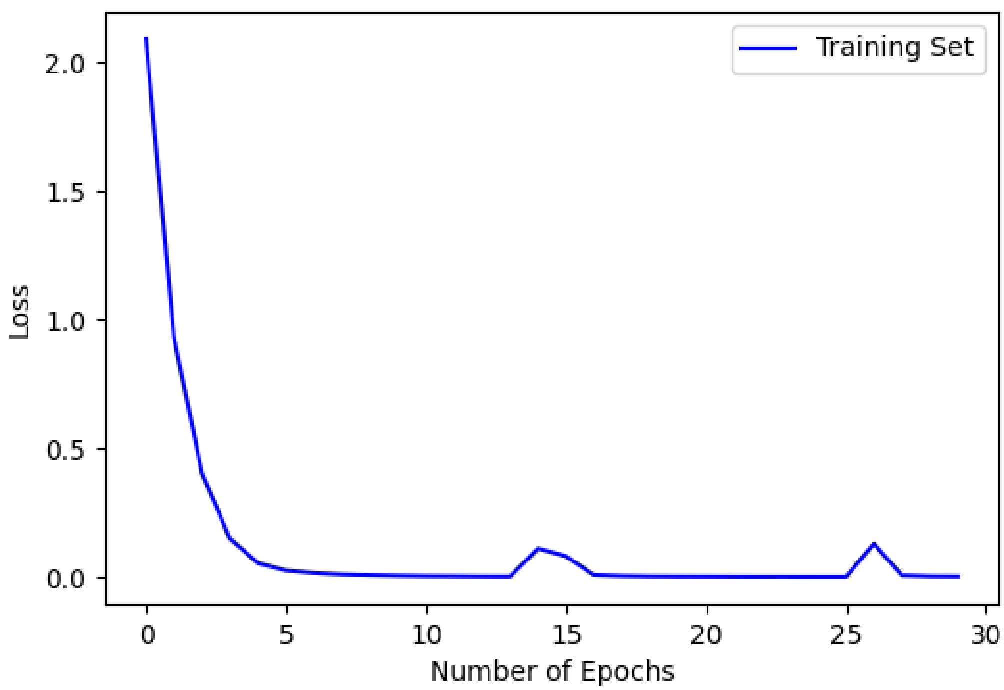


Figure 3: Training Loss for 30 epochs with 20 classes. The network is modified with an additional Conv2D layer and batch normalization.

The overall classification accuracy for this modified network is 62%

**Q2**

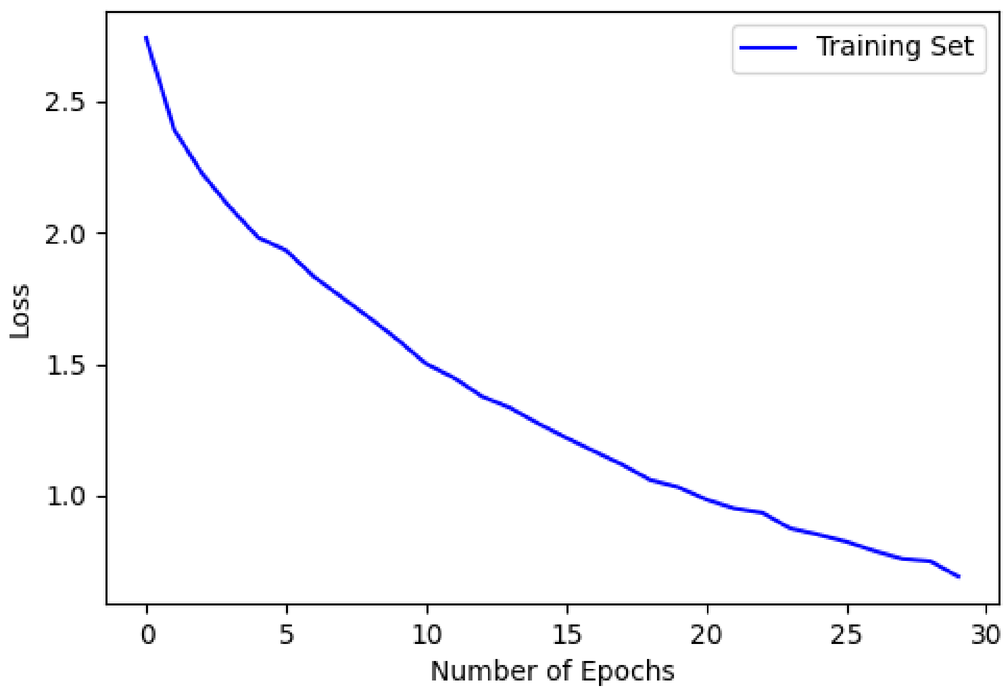


Figure 4: Training Loss for 30 epochs with 20 classes. The network is modified with an additional Conv2D layer, batch normalization, and dropout.

The overall classification accuracy is 45%.

## Training Neural Network with Validation

### Q1

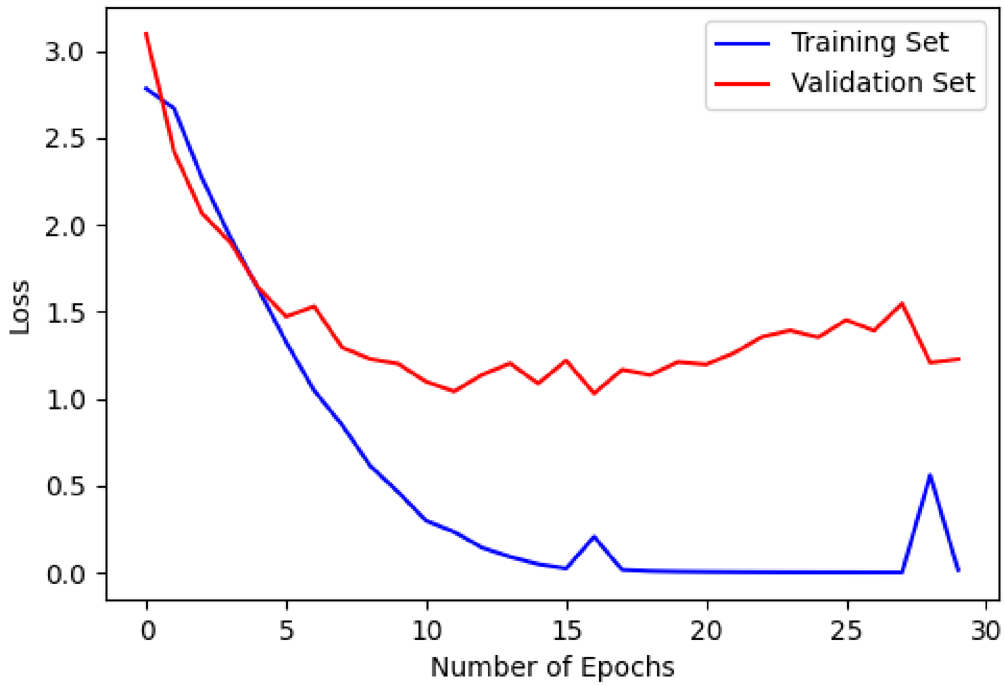


Figure 5: Training and Validation Loss for 30 epochs with 20 classes. The network is modified with an additional Conv2D layer, batch normalization, and dropout.

### Q2

The overall classification accuracy is 70%.

## Hyperparameter Tuning

### Q1

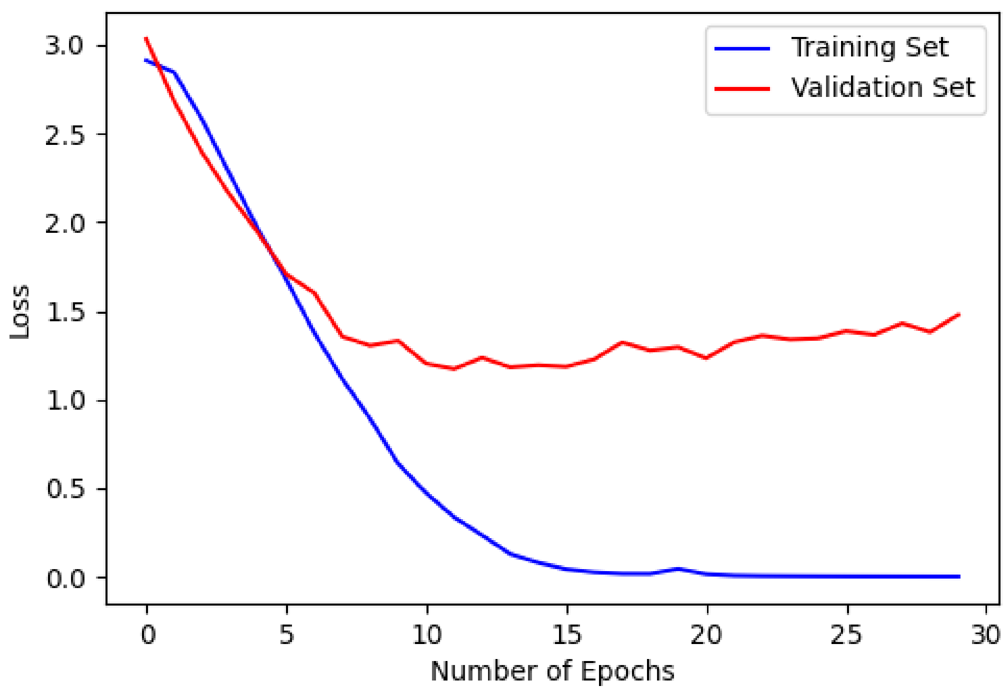


Figure 6: Training and Validation Loss for 30 epochs with 20 classes. The optimizer used is Adam

The overall classification accuracy is 55%.

**Q2**

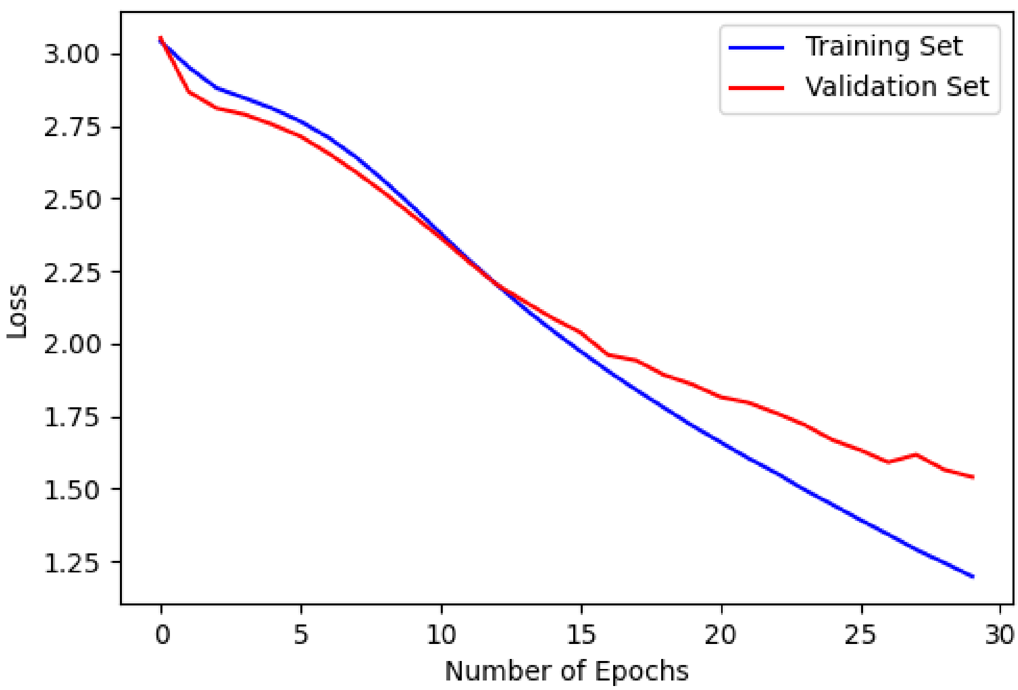


Figure 7: Training and Validation Loss for 30 epochs with 20 classes. The optimizer used is RMSProp as we got the best results from RMSProp. The learning rate is 1e-5

In Fig. 7, we can see that the training and validation loss curves still tend to decrease and have not settled even after 30 epochs. Due to the lower learning rate, smaller steps are taken in the gradient direction, thus resulting in slower convergence. In the case of low learning rate, one solution could be to increase the number of epochs to iterate until the loss curves become stable.

The overall accuracy is 55%.

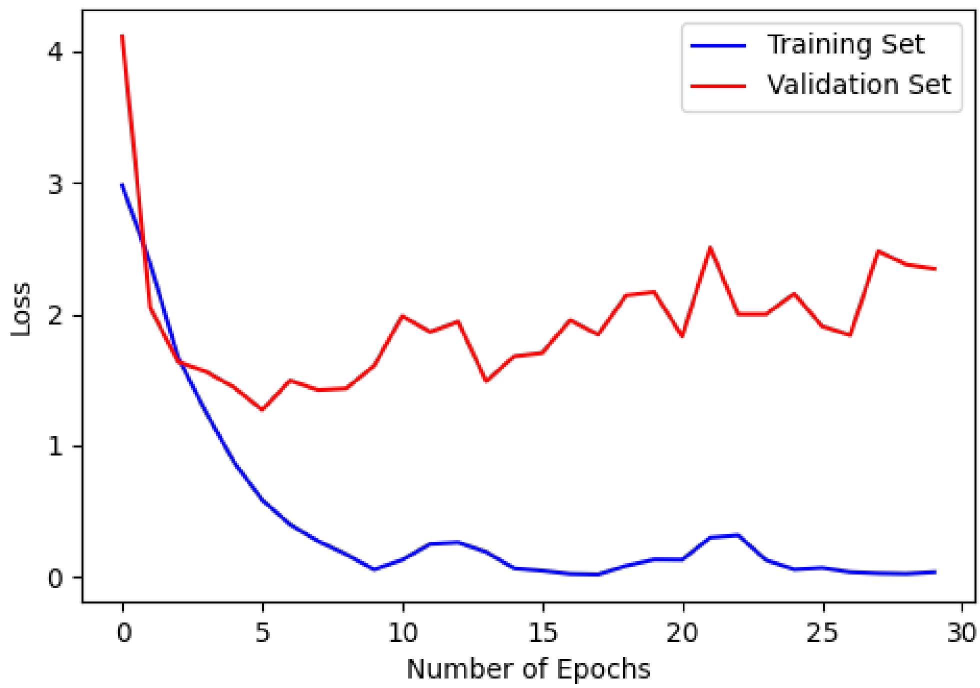


Figure 8: Training and Validation Loss for 30 epochs with 20 classes. The optimizer used is RMSProp. The learning rate is 1e-3

In Fig. 8, we can see that the training and validation loss curves settle faster than Fig. 7. Due to the higher learning rate, larger steps are taken in the gradient direction, but this may not always lead to the optimum solution.
The overall accuracy here is 63%.

**Q3**

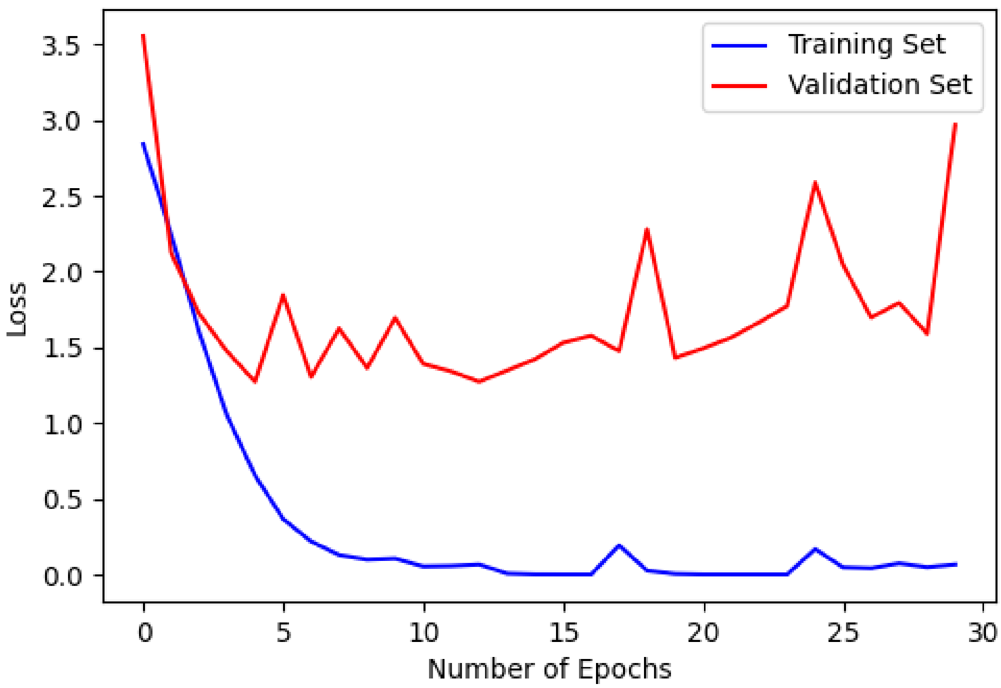


Figure 9: Training and Validation Loss for 30 epochs with 20 classes. The optimizer used is RMSProp. The batch size is 8

The overall classification accuracy for batch size of 8 is 71%.

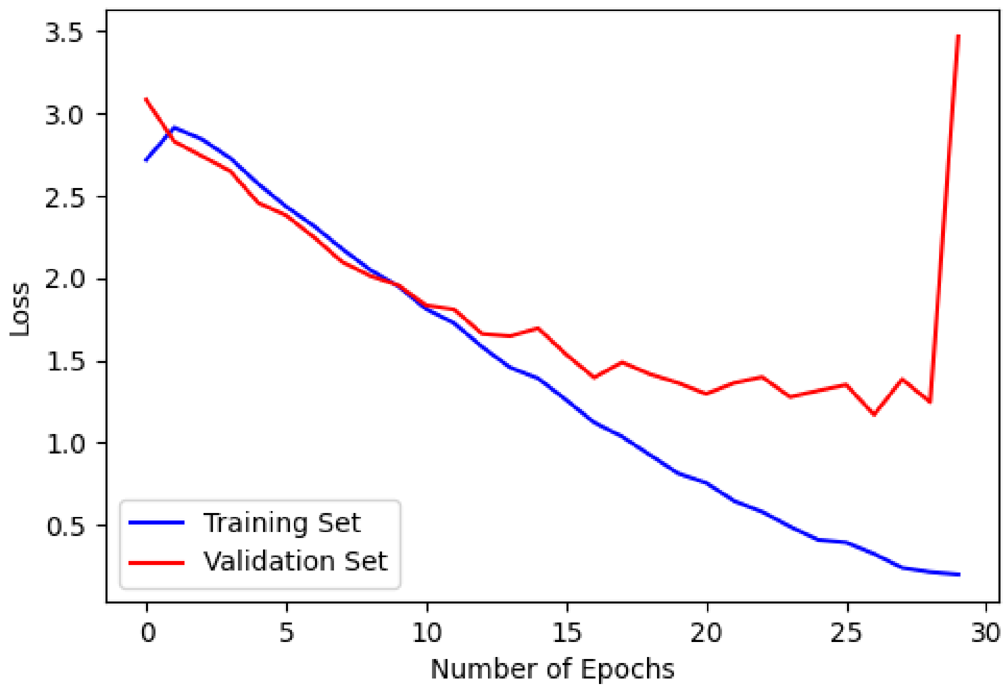


Figure 10: Training and Validation Loss for 30 epochs with 20 classes. The optimizer used is RMSProp. The batch size is 64

The overall classification accuracy for the batch size of 64 is 57%.

Having a smaller batch size makes the optimizer take smaller steps in the gradient direction as compared to having a larger batch size. The advantage of having a large batch size means we may have more confidence in the gradient caluclation and thus can take larger steps, however, this may lead to getting stuck in a local minima. On the other hand, such bouncing due to small batch sizes may result in unstable propagation, but can also help in getting out of a local minimum. In this example we can see that the model performs better when the batch size is lower.