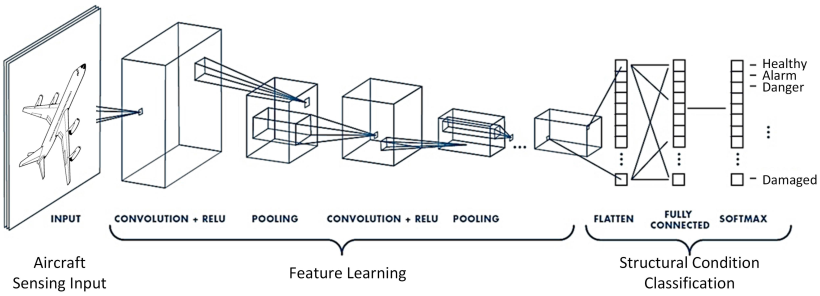
AML Homework 2

Convolutional Neural Network



Debodeep Banerjee
Trina Sahoo
Paolo Falcone
Dario Baraghini

November 2021

Question 1.a

We have implemented the conv2D layer with kernel size 3, maxpool with stride 2 and the Relu activation function.

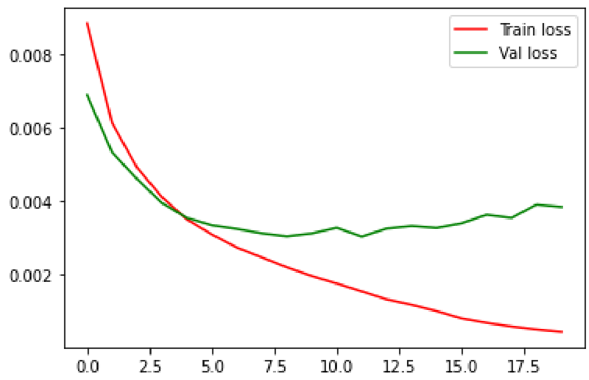


Figure 1: Train and validation loss curve

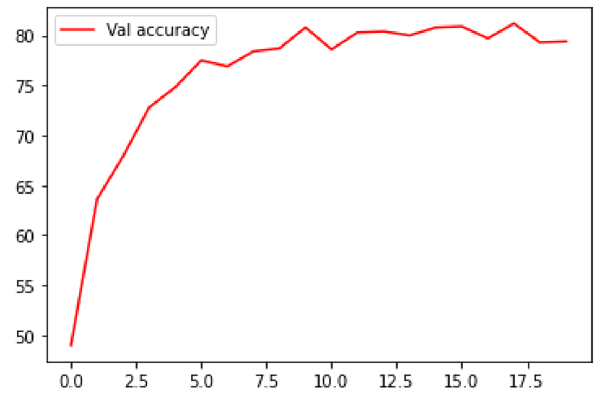


Figure 2: Validation accuracy curve

After 20 epochs, the validation accuracy reached 79.5%.

Question 1.b

The total number of parameters in the model is 7678474 and after adding the BatchNormalization and dropout layers it changed to 7682826.

Question 1.c

The following two images depicts the visualization of the filters before and after the training with 20 epochs.

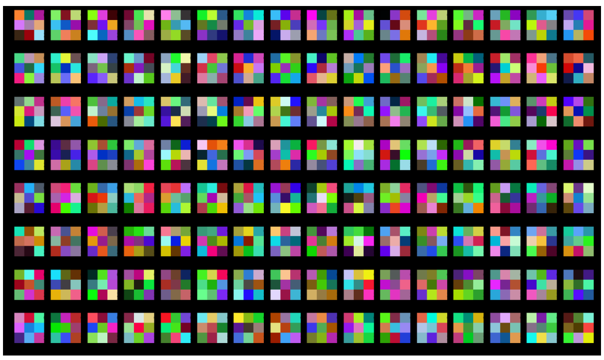


Figure 3: before training

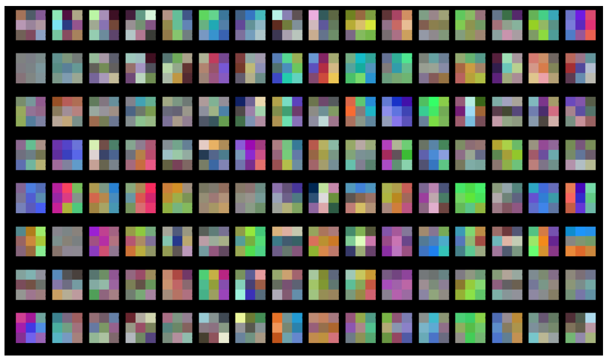


Figure 4: after training

As we can see in figure 3, the organization of the colored boxes is quite random and after training with 20 epochs, the colored boxes are tending to be clutched amongst themselves. We expect to improve the learning process if we increase the number of epochs(to a certain extent).

Question 2.a

Batch Normalization:
Batch normaliation is one of the techniques to prevent the model from over fitting. In this solution, we have implemented the batch normalization given the other parameters fixed. After 20 epochs we could observe an improvement of the validation accuracy from 79.5% to 82.7%. If we compare figure 6

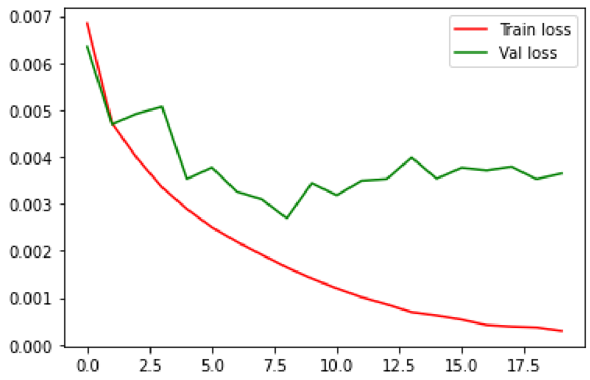


Figure 5: before training

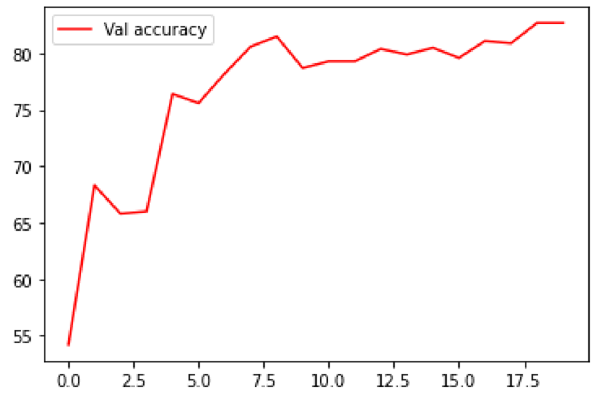


Figure 6: after training

and figure 2, we could see that the validation accuracy improved after using the batch normalization.

Question 2.b

Early Stoppage:
For the Early Stoppage implementation we added a small but effective code during the training, on each iteration we checked if the accuracy was improving or not, if for N iterations (where N is our patience) it wouldn't improve we would break the for loop.
Here we will show the results that we got from running with 40 epochs, with batch normalization and without it, also changing the patience value.

Best Epoch	Stoppage Epoch	Best Accuracy	Stoppage Accuracy	Patience
10	15	82.8%	80.8%	5
9	14	81.6%	80.7%	5
5	10	79.9%	79.2%	5
9	14	82.0%	79.8%	5
12	17	82.4%	80.9%	5
2	12	80.9%	80.5%	10
34	NA	84.5%	80.3%	10

Table 1: Application of early stopping using batch normalization.

As we can see from the results provided using the Batch Normalization the Early Stoppage was helping the model avoid overfitting, in fact in the first 10-15 epochs usually it was getting to a very good accuracy and if it kept running for all of the 40 epochs it would probably have performed worse on the test.
When the Patience was = 10 we saw much different results. In fact the first run was very short, luckily

scoring 80.9 at the second epoch, instead in the second run it ran until epoch 40. These results show how a different patience can lead to higher accuracy but also overfitting if it's a too high value.

ES_Epoch	Last_Epoch	ES_Accuracy	Last_accuracy	Patience
15	20	80.2%	79.6%	5
26	32	81.2%	79.7%	5
11	16	80.7%	79.5%	5
16	19	80.2%	78.3%	3
10	13	80.4%	80.1%	3
22	34	81.9%	78.8%	10
24	34	81.6%	79.8%	10
19	29	80.8%	79.6%	10

Table 2: Application of early stopping without using batch normalization.

Question 3.a

In this question we have checked all of the combinations in data augmentation like translation, rotation, scaling, clipping, random cropping, color transformations and also greyscaling and color-jittering. The combination we used is: horizontal flip with p=0.5, random perceptive with distortion scale = 0.5, p = 0.5, interpolation = 3, fill = 0 and color-jitter with brightness = 0.1, saturation = 0.1, contrast = 0.05 and hue = 0.05. This got us 86.1% as the final validation accuracy which was the highest. The loss and accuracy curves obtained are shown below:

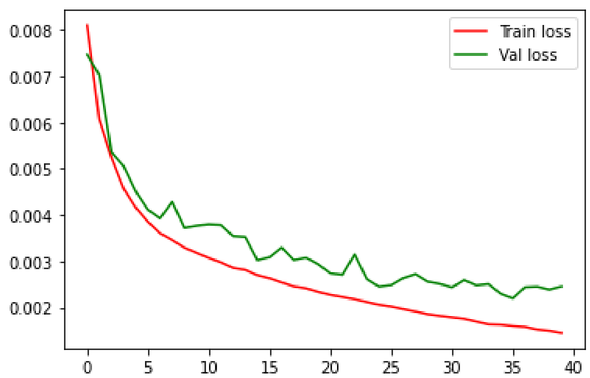


Figure 7: Train and validation loss curve

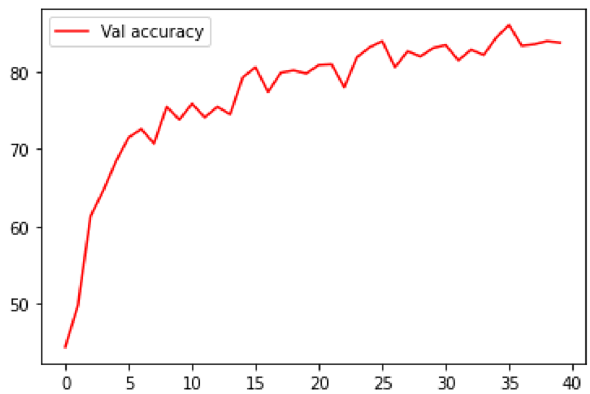


Figure 8: Validation accuracy curve

Question 3.b

By adding the dropout layer with p = 0.1 at every layer we have the highest validation accuracy without using data augmentation. This probability p also provides the least training loss and least validation loss. The highest test accuracy obtained with the probability value is 81.3%. The train loss, valid loss and validation accuracy plots with different probability values for dropout are shown below:

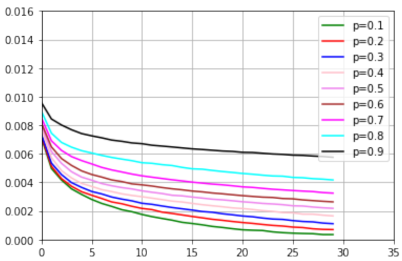


Figure 9: Train loss curve

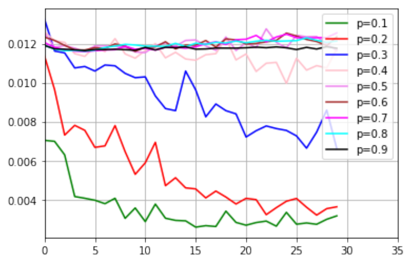


Figure 10: Validation loss curve

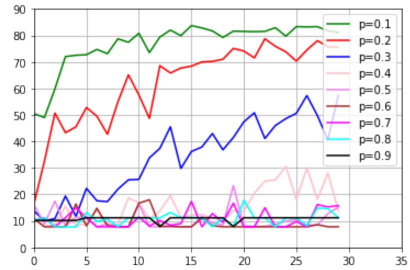


Figure 11: Validation accuracy curve

With the data augmentation, by adding the dropout with probability p = 0.1 at every layer gives us the highest accuracy. This probability p also provides the least training loss and least validation loss. The highest test accuracy obtained with the probability value is 85.04% which is greater than the test accuracy obtained without data augmentation.The train loss, valid loss and validation accuracy plots with different probability values for dropout after data augmentation are shown below:

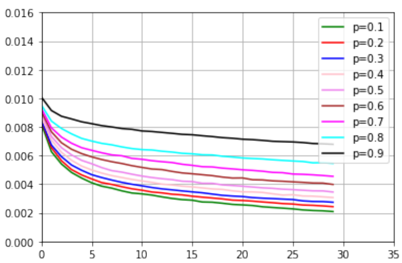


Figure 12: Train loss

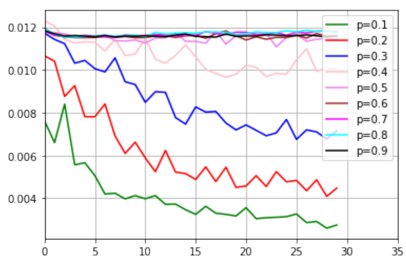


Figure 13: Validation loss

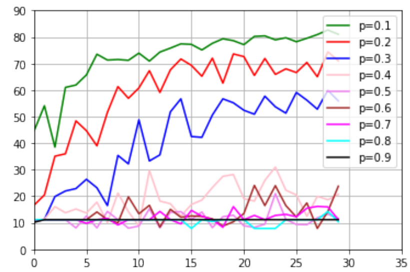


Figure 14: Validation accuracy

As the dropout increases for both the cases, the performances decreases. After p = 0.4, the performance reduced significantly. The training loss and the validation loss increases and the validation accuracy falls.

Question 4.a

In this part we are using `vgg11_bn` as a pretrained model. The validation accuracy and test accuracy are 64.6% and 64.1% respectively. The training loss and validation loss curves and also the validation accuracy are provided below:

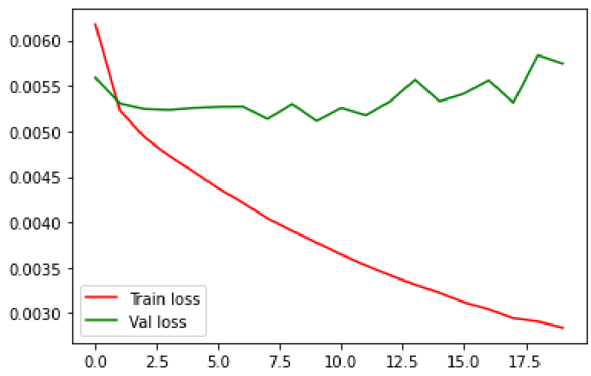


Figure 15: Train loss and val loss

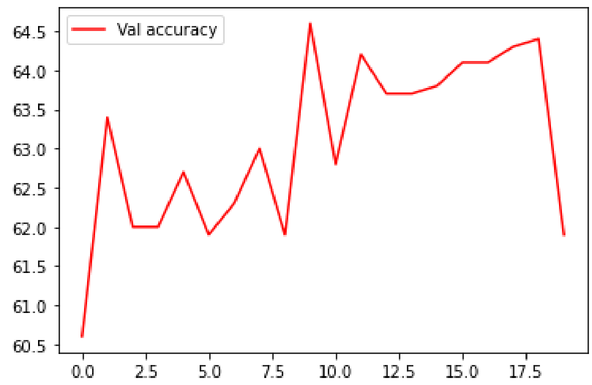


Figure 16: Validation accuracy curve

Question 4.b

For the fine tuning part, we have made the following changes:

1. The `fine_tune` and pretrained is set as false.
2. The learning rate delay has been changed to 0.95.
3. In the data augmentation both RandomHorizontalFlip with $p = 0.5$ and RandomGrayScale with $p = 0.5$ has been used.
4. For the test accuracy, the model with the highest has been used.

The validation accuracy and the test accuracy have increased to 90.7% and 89.1% respectively. The training loss and validation loss curves and also the validation accuracy are provided below:

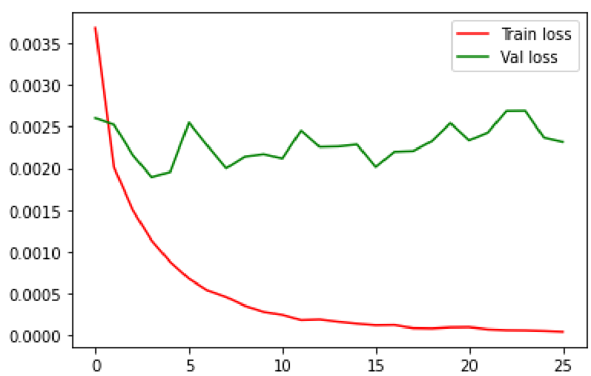


Figure 17: Train loss and val loss

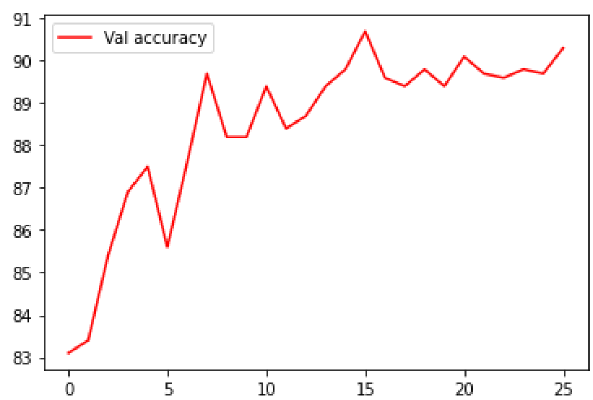


Figure 18: Validation accuracy curve