**CONVOLUTION NEURAL NETWORK**

A particular kind of machine learning model called a convolution neural network is created to learn from examples, or training samples, in order to predict or decide on new, unobserved data. A neural network is trained to recognize patterns of images and relationships in the training data, which it may use to generalize and produce precise predictions on un trained data. It has convolution layers that reduce spatial size for features. The neural network's performance can be significantly impacted by the architecture that is selected, especially when the size and capability of the training dataset are taken into account.

In general, more training samples are required for a neural network to be efficiently trained the more complicated the problem or task. The network architecture would probably need to be much bigger and more complex if you were trying to train a neural network to recognize more complicated items, like cars or animals. This is due to the fact that there are a lot more potential aspects and variances in how cars and animals look, and the network would need to be able to learn a considerably wider variety of characteristics and trends in order to correctly recognize them. We will work on the following data sets to identify whether they had any relation between sample size and choice of network.

1.Train=1000, Test=500, Validation=500 (optimizer=RMS)

a. In this accuracy of the model is highest () and the loss is minimum at out of epochs. As such both metrics are affected afte .

b. validation accuracy has not shown improvement at epoch 16 training loss decreasing to certain point with accuracy 0.91 and validation accuracy up to 0.72 after using dropout techniques at epoch43 validation accuracy is 80%.

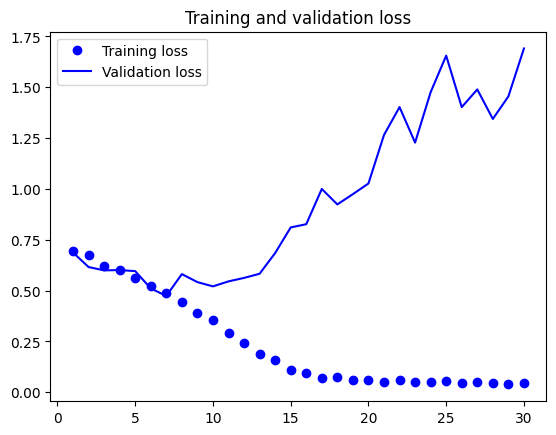
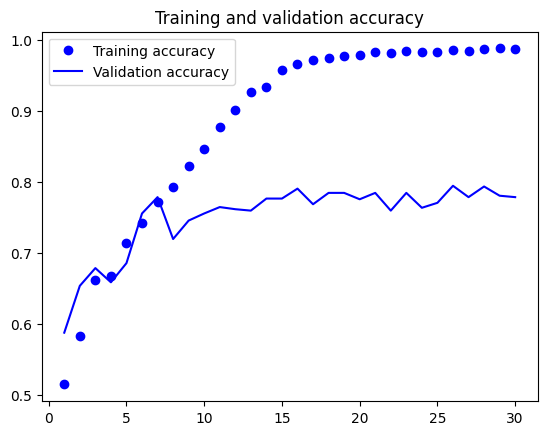
Training sample=1000, Validation sample=500, Test sample=500 and Optimiser=RMS

|  |  |  |
| --- | --- | --- |
| ACCURACY | SCRATCH MODEL | AUGUMENTATION  DROPOUT |
| VALIDATION SET | 72 | 80 |
| TEST SET | 72 | 84 |

Training sample=2000, Test=500, Validation=500 from graph at epoch=7 looks like an optimal value at this point the model starts overfitting. After applying augmentation technique at epoch 39 model test accuracy decreased than validation accuracy.

|  |  |  |
| --- | --- | --- |
| ACCURACY | SCRATCH MODEL | AUGUMENTATION  DROPOUT |
| VALIDATION SET | 77 | 88 |
| TEST SET | 73.9 | 86.9 |

The following set sample2000 of training data with graphs shows where the model was going to over fit at some point to choose optimal value for the epoch to run the test set



From above sample we found that increasing training sample will result in increasing the accuracy of the model. Now we also using the pre trained model along with scratch model to check whether accuracy increasing or not

Training sample=3000, Test=500, Validation=500 from graph at epoch=7 looks like an optimal value at this point the model starts overfitting. After applying augmentation technique at epoch 27 model test accuracy decreased than validation accuracy. Similarly, in pre-trained model at epoch2 validation losses started increasing I found it is an optimal value with same accuracy in pre trained model accuracy increased.

|  |  |  |  |
| --- | --- | --- | --- |
| ACCURACY | SCRATCH MODEL | AUGUMENTATION  DROPOUT | PRETRAINED  MODEL |
| VALIDATION SET | 77 | 85 | 98 |
| TEST SET | 78.6 | 90 | 97.5 |

Training sample 4000 also at same point of epoch overfitting has occurred

|  |  |  |  |
| --- | --- | --- | --- |
| ACCURACY | SCRATCH MODEL | AUGUMENTATION  DROPOUT | PRETRAINED  MODEL |
| VALIDATION SET | 83 | 81.8 | 97 |
| TEST SET | 80.5 | 82.4 | 97.9 |

Training sample 5000 scratch model epoch 8 pre-trained model at epoch 2 optimal value of validation accuracy

|  |  |  |  |
| --- | --- | --- | --- |
| ACCURACY | SCRATCH MODEL | AUGUMENTATION  DROPOUT | PRETRAINED  MODEL |
| VALIDATION SET | 85 | 80 | 97.2 |
| TEST SET | 87.5 | 79.8 | 97.9 |

Training sample=8000, Test=500, Validation=500 from graph at epoch=7 looks like an optimal value at this point the model starts overfitting. Similarly, in pre-trained model at epoch2 both losses started increasing I found it is an optimal value with same accuracy in pre trained model accuracy increased.

|  |  |  |  |
| --- | --- | --- | --- |
| ACCURACY | SCRATCH MODEL | AUGUMENTATION  DROPOUT | PRETRAINED  MODEL |
| VALIDATION SET | 90 | 86 | 98 |
| TEST SET | 89.2 | 86 | 98.8 |

**Conclusion**

As the above accuracy results every sample size is effecting the accuracy of the model and when we applied the augmentation and regularization techniques there is significant increase in the accuracy for 3000 sample set. Then we tried to increase the sample size for 4000, 5000 and 8000 at this samples accuracy with regularization methods reduced when compared with scratch model accuracy. But with same sample sizes given highest accuracy in pre trained model to 98.8 on test set.

we can say the sample size and complexity of the training data should be taken into account while selecting the network architecture. A larger dataset might need a network with a complex architecture for identifying the underlying patterns, but a smaller dataset might only need a more basic architecture to avoid overfitting on small data set. If we consider larger data set, we can train more complex data. If we use low sample size model performance on unseen data will be poor.

The choice of network architecture and the relationship between the training samples is the right balance that needs significant consideration and evaluation. While avoiding over fitting or under fitting, a well-designed neural network architecture should be able to successfully simulate the underlying patterns in the training data.