Assignment:3

Convolution networks (convnets)

For this assignment our limited dataset originates from the “Dog-vs-Cats” dataset on the Kaggle website. Our network is a convolution network, or convnet, that is a particular type of deep learning model suited for computer vision and that, thanks to its properties, provide reasonable results even when it is trained with a limited amount of data.

1. Consider the Cats & Dogs example. Start initially with a training sample of 1000, a validation sample of 500, and a test sample of 500 (like in the text. Use any technique to reduce overfitting and improve performance in developing a network that you train from scratch. What performance did you achieve?

Test Accuracy – 72.80%

Training Accuracy - 97.85%

Validation Accuracy – 77.00%

So, we got the above results when we used a training sample of 1000 images and using dropout to reduce overfitting.

2. Increase your training sample size. You may pick any amount. Keep the validation and test samples the same as above. Optimize your network (again training from scratch). What performance did you achieve?

Test Accuracy - 82.60%

Training Accuracy – 88.35%

Validation Accuracy – 82.20%

So, there is a high increase in Train, Validation and Test (From 72.8% to 82.60%) accuracy when the number of training samples is increased, and data augmentation is used as expected.

3. Now change your training sample so that you achieve better performance than those from Steps 1 and 2. This sample size may be larger, or smaller than those in the previous steps. The objective is to find the ideal training sample size to get the best prediction results. Increase the Training sample size to 2000 and monitor the results.

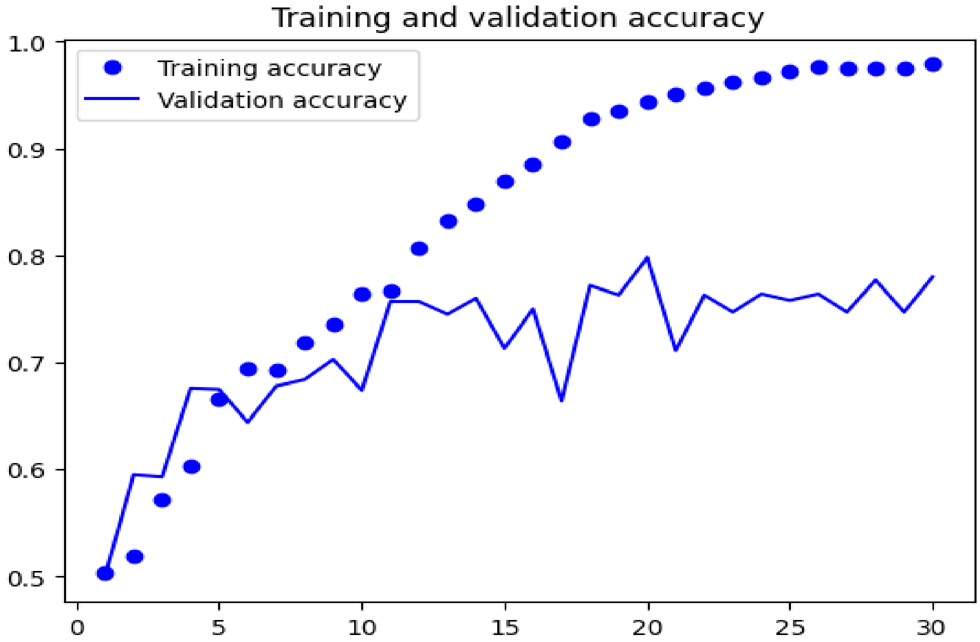
Test Accuracy – 76.00%

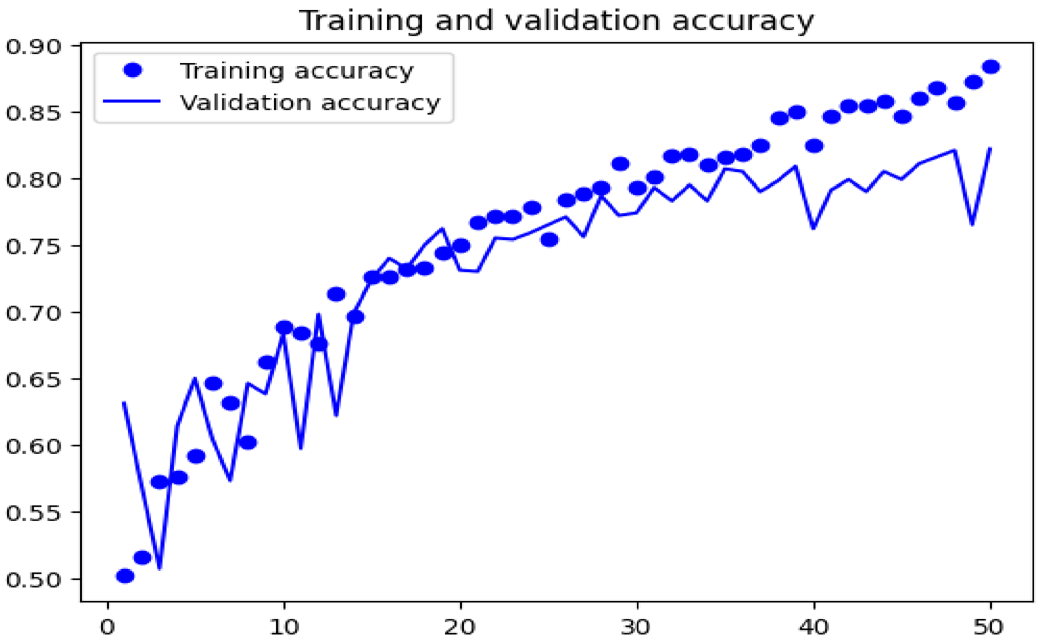
Training Accuracy – 80.70%

Validation Accuracy - 78.30%

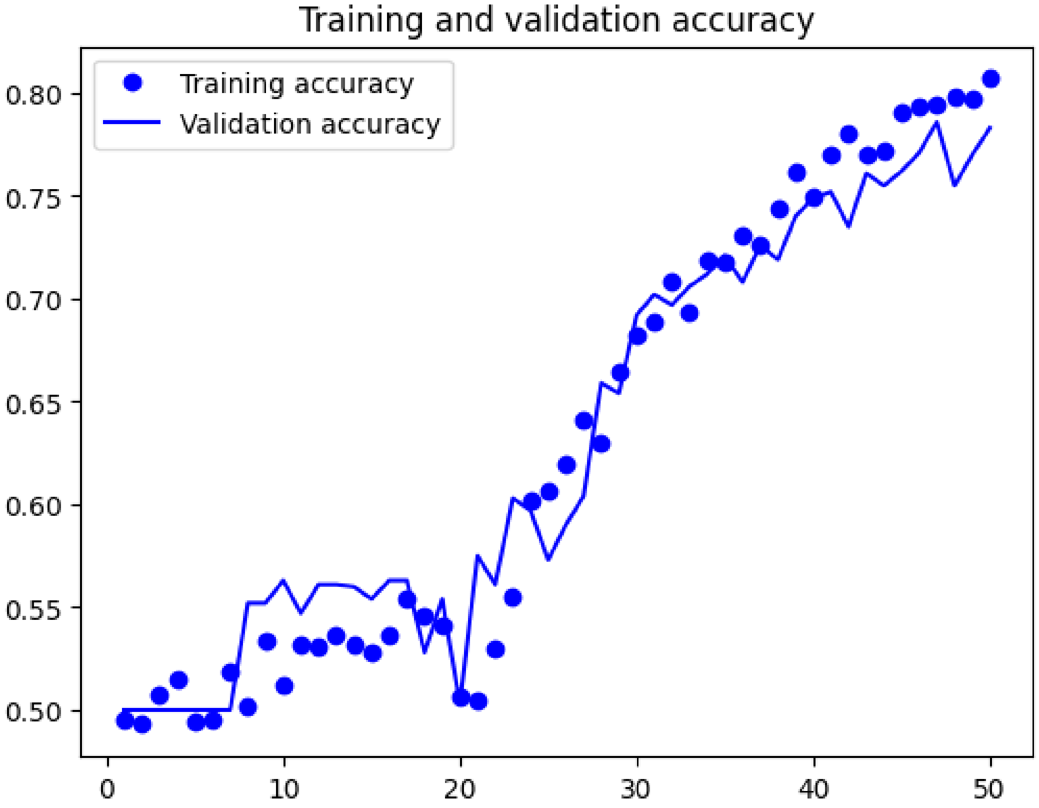
Model	Training Accuracy	Validation Accuracy	Test Accuracy
Training 1000 validation 500 Test 500	0.978	0.780	0.728
Training 1500 validation 500 Test 500	0.883	0.822	0.826
Training 2000 validation 500 Test 500	0.8070	0.7830	0.763

Below are the plots of Training and validation accuracy:





Training and validation accuracy



Training and validation accuracy

The three models that were built from scratch have shown that the accuracy of the model can be increased by increasing the size of the training sample.

By changing the training sample, optimization techniques can be utilized to avoid overfitting. A different approach to making the most of training data is data augmentation.

Reducing the model's size or the number of learnable parameters can significantly reduce overfitting. Restricting the weights to small values can also regulate the distribution of weight values and decrease the complexity of the network.

The model can incorporate optimization approaches like learning rate and dropout. Compared to the old model, the new model had greater accuracy because it used optimization techniques and a larger training sample size.

4. Repeat Steps 1-3, but now using a pretrained network. The sample sizes you use in Steps 2 and 3 for the pretrained network may be the same or different from those using the network where you trained from scratch. Again, use all optimization techniques to get best performance.

Test Accuracy – 81.40%

The following are some conclusions that can be drawn from using a pre-trained VGG16 convnet network compared to a model trained from scratch.

In summary, using a pre-trained model for various image recognition tasks has several advantages. Firstly, it requires less training and effort in designing the model's architecture. Secondly, using a pre-trained model yields significantly higher accuracy compared to a custom-built convolutional neural network (CNN). Thirdly, we observe that a large sample size enhances the model's learning ability when training from scratch. However, in a pre-trained model, we can improve the results by unfreezing some of the top layers of a frozen model and jointly training both the newly added part and the pre-trained layers.

Conclusion

Observed that the performance of the models we created is influenced by various factors, with the choice between training a model from scratch or using a pretrained model having the greatest impact. Even with optimal model configurations, it is unlikely to achieve the accuracy of a pretrained network.

However, we were able to attain a decent performance with a small amount of training data. The largest impact on performance was from a larger training set rather than a larger validation set. We can experiment with other partitions and increase epochs to further improve the network, and modifying the network structure can potentially enhance performance.

Data augmentation helped improve accuracy for the first data partition, especially when the training set was small. As the training set increased, data augmentation became less significant in improving the model's accuracy.