ASSIGNMENT 2: CONVOLUTION REPORT

INTRODUCTION:

The goal of this assignment is to develop a convolutional neural network primarily for applications in computer vision from scratch level. The dataset we are utilizing is the "Dog-vs-Cats" dataset on Kaggle. The task of creating an effective model is difficult due to the small amount of data available to us. Convolutional neural networks, also referred to as convnets, are common deep learning models that have shown to be quite successful in computer vision applications. One of Convnet's primary benefits is its ability to detect and identify visual trends in images. As a result, they succeed in tasks like image recognition, object identification, and segmentation.

We will look at the connection between training samples and whether we should train our model from scratch or by utilizing a pretrained convnet in this assignment.

Initially, we are dealing with the sample of 1000 data points for training and 500 for validation and 500 for test then we measure the performance of the model built. Then, we increase the size of our training dataset by 1000 keeping the validation and test dataset of same size and measure the performance. Again, we increase the size of the training dataset by 3000 and measure the performance, by doing so the ideal goal is to find the appropriate training dataset size for best predictions. Moreover, we use the pretrained model on each dataset size mentioned above and measure the performance to know the difference between the models and to get the model with best performance.

Let’s look at the summary of the results obtained by performing the above tasks on the data set and formulate the results for better understanding:

SUMMARY:

Q1. 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?

Starting initially with training sample size of 1000, validation of 500 and test of 500, we build the model from scratch with and without any regularization techniques and note the differences in training, validation, and test accuracy also the validation loss.

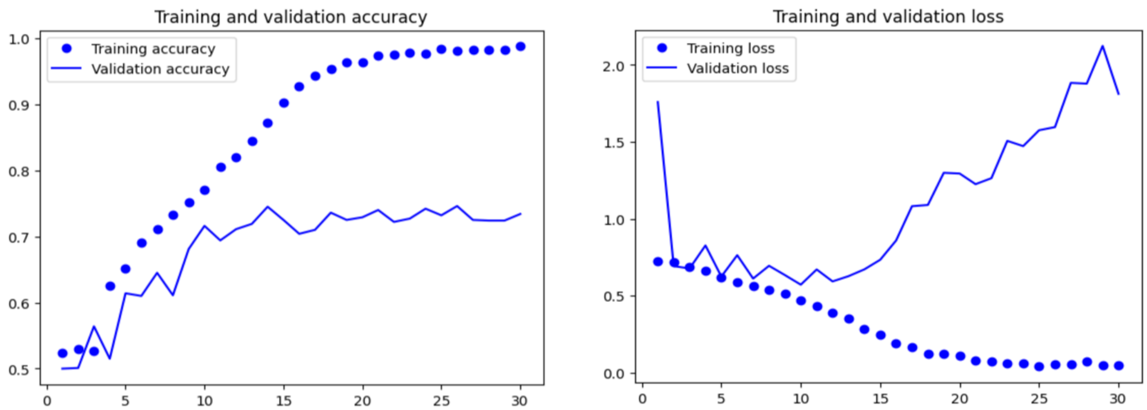

Fig 1: Without Regularization of the model

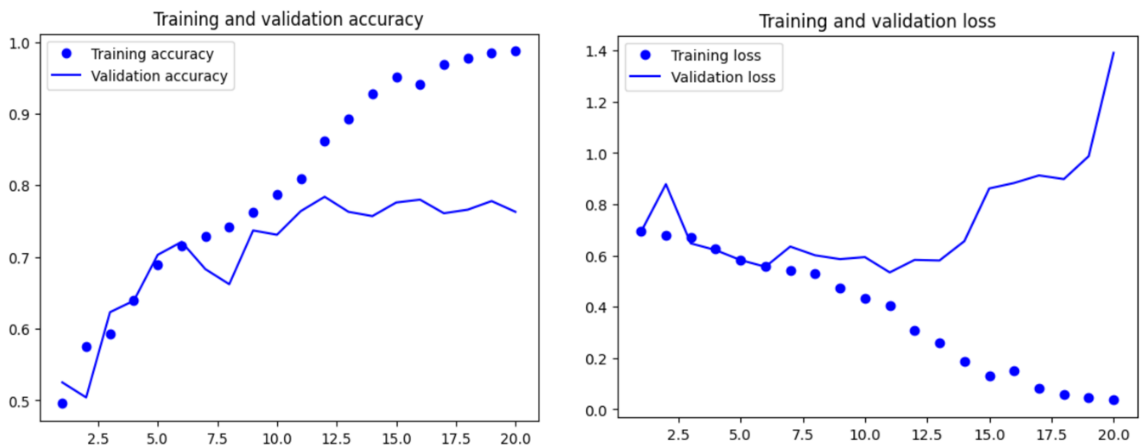

Fig 2: With Regularization technique used for the model

The Figure 1 shown above are instances of overfitting. Validation accuracy stalls at 72-75%, whereas training accuracy rises linearly over time, until it reaches nearly 100%.

And the validation loss reaches its minimum after 7 epochs and then stalls, whereas the training loss keeps decreasing linearly until it reaches to 0.

Our primary problem will be overfitting because there are a smaller number of training samples. Several strategies, including dropout, regularization, and data augmentation, can be used to reduce overfitting.

As mentioned we can use any technique to enhance the model's performance, I used **dropout regularization** technique to reduce overfitting and improve performance.

Let's Summarize the results,

MODEL	TEST ACCURACY	VALIDATION ACCURACY	TEST LOSS
Without regularization	70%	73%	0.6
With dropout regularization	73%	78%	0.59

According to the results, the performance of the model improves after regularizing the model using dropout technique. The accuracy increases from 70 to 73 and there is a little decrease in test loss from 0.6 to 0.59.

Let's now increase the sample size by 1000 and see the change in performance of the model.

Q2. 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?

Now, I have increased the sample size of training by 1000 keeping the validation and test size the same, the new size is 2000 and I used data augmentation technique and dropout technique for regularization.

MODEL	TEST ACCURACY	VALIDATION ACCURACY	TEST LOSS
With 1000 samples	73%	78%	0.59
With 2000 samples	85%	88%	0.34

We can see that the performance of model has now greatly increased by increasing the training sample size and using data augmentation as regularization technique, the test accuracy has changed from 73% to 85% whereas the test loss has decreased to 0.34 from 0.59.

Q3. Now change your training sample so that you achieve better performance than those from Steps1 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 best prediction results. ?

Now, I have increased the training sample size from 2000 to 5000 keeping the validation and test sample size the same. Let's summarize the results.

MODEL	TEST ACCURACY	VALIDATION ACCURACY	TEST LOSS
With 1000 samples	73%	76%	0.59
With 2000 samples	85%	88%	0.34
With 5000 samples	91%	92%	0.19

There is an increase in test accuracy of 91% from 85% with increase in training samples from 2000 to 5000 whereas there is decrease in test loss from 0.34 to 0.19 which means our model performance has been improving with increase in the training sample data size.

Therefore, as the sample size increases the performance of the model increases resulting in better accuracy and we can say for the cat's vs dogs example, the regularized model with 5000 samples is best for getting best predictions, it can be considered as the ideal training dataset size.

Q4. 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 any and all optimization techniques to get best performance.

As a next step, to improve the accuracy of the model, we will use pretrained model, I am using the VGG16 pretrained convolution network as the base network in fine tuning process.

Let's summarize the results obtained by fine tuning the models with different sample size of training data sets.

FINE TUNED MODEL	TEST ACCURACY	VALIDATION ACCURACY	TEST LOSS
With VGG16 pretrained model	71%	73%	0.56
VGG16 and 1000 samples	97%	97%	0.16
VGG16 and 2000 samples	99%	98%	0.02
VGG16 and 5000 samples	99%	99%	0.06

From the above table we can see that the testing and validation accuracy tend to improve and the validation loss goes down as the sample size increases. They are even better when regularized and fine-tuned when compared to only regularized models.

In Summary, When the training data set is small, overfitting becomes the main concern. A set of experiments looked at how varying training sample sizes, data augmentation, and optimizer choices affected model performance. Remarkably, when the training size was raised from 1000 to 5000 samples, both accuracy and loss greatly improved. This highlights how important having ample training data is to improve the performance of the model. Furthermore, the models trained on augmented data outperformed the ones not trained on it. Data augmentation is an efficient way to increase the effective size of the dataset and improve the model's ability to generalize from small amounts of data. **In the end,** the pretrained, data-augmentation-enabled model with the highest accuracy was trained on a 2000 and 5000 sample set. The results highlight the significance of the choice, data augmentation, and size in model training and show how these elements work together to provide better model performance.

SUGGESTIONS FOR BEST PREDICTIONS:

- Pre-trained convolutional neural networks developed on the ImageNet dataset greatly enhanced the model's performance; larger training sample sizes allowed for the extraction of more features and helped reduce overfitting.
- The model's capabilities greatly improved accuracy with pre-trained CNNs and data augmentation, particularly with smaller training sample sizes. This suggests that employing pre-trained models can be beneficial even with a limited amount of training data.