**CONVOLUTION REPORT**

**DINESH YECHURI**

**811295667**

The goal of this project is to build a convolutional neural network that can reliably and accurately recognize images of cats and dogs by identifying the unique features of each animal.

In this experiment, a Kaggle dataset of 12,500 test and 25,000 training images the same number of dogs and cats—is employed.

**Problem to be defined:**

The goal of the Cats-vs-Dogs dataset is to identify if an image is of a dog or a cat.

**Methods**

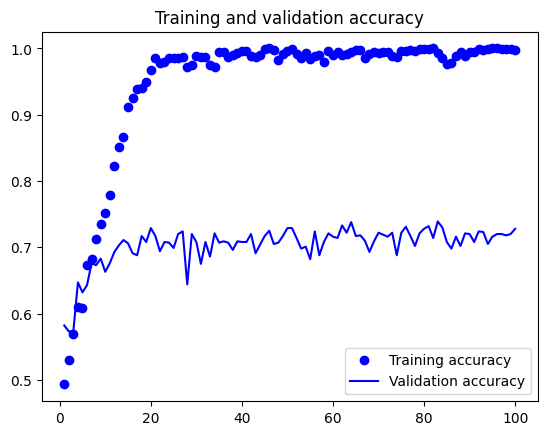
**Data:**

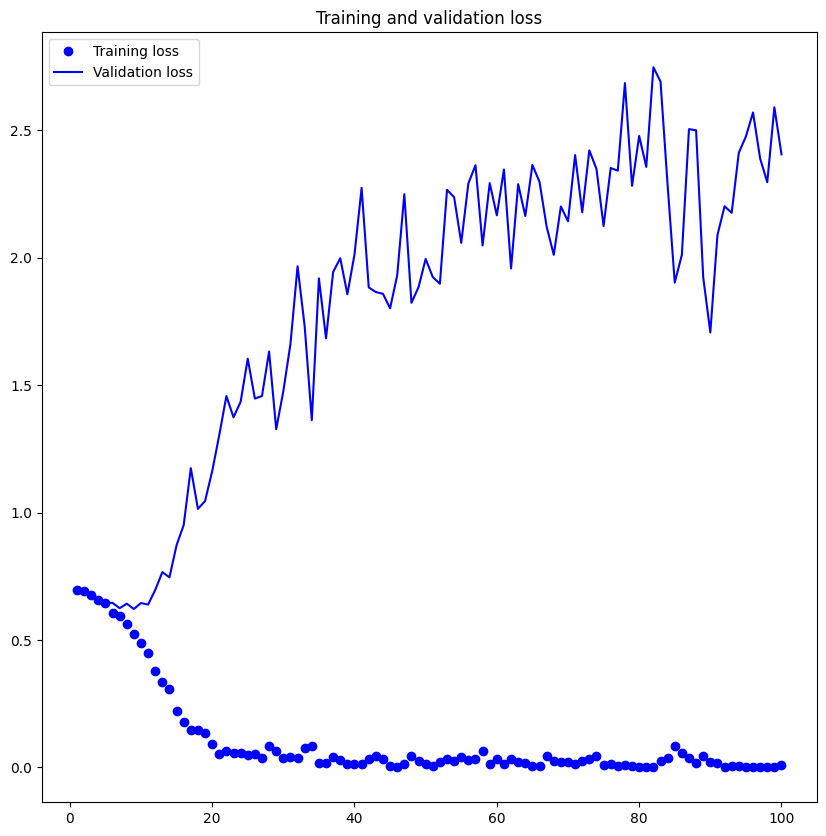
The Cats-vs-Dogs dataset, which has a compressed size of 543MB, has 25,000 images that are split equally between dogs and cats. After downloading and unzipping the dataset, I created a new dataset consisting of three subsets:   
• A training dataset of one thousand samples per class.   
• 500 sample validation dataset   
• 500 samples in the test dataset   
We need to increase our neural network because the problem we are working on is more difficult and requires a broader image. We will incorporate a step into our current Conv2D + MaxPooling2D design to accomplish this. This not only increases the network's capacity but also ensures that the feature maps won't be too large when we get to the Flatten layer. At first, the dimensions of our input images are 150 x 150. The feature maps get smaller as we move up the network's stages, reaching 7x7 just before the Flatten layer. Despite appearing somewhat random, the input size selection is appropriate for the task at hand.

**Data Preprocessing:**

• Check the picture files.   
•Transform the JPEG data into pixel grids that are RGB.   
• Construct tensors using floating points using grids.   
  
Since neural networks function best with small input values, the pixel values, which vary from 0 to 255, should be rescaled to the [0, 1] interval.’

I converted the data transformation using the data flattening approach, taking 255 as the batch size. We were able to determine the test accuracy to be 70.3 % and the validation accuracy to be 72.8% with the aid of hundred epochs.



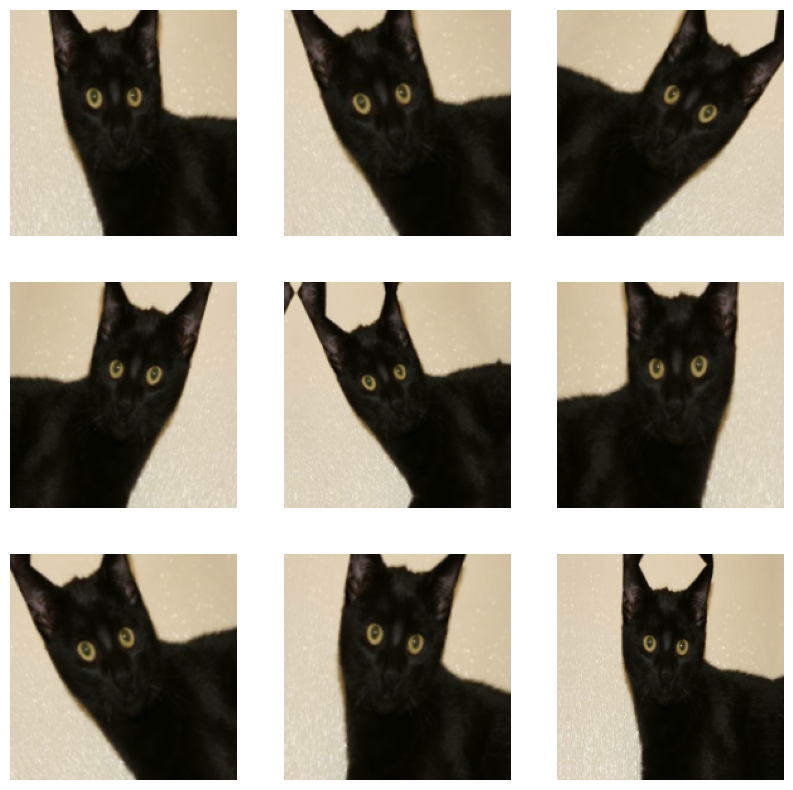


Based on the aforementioned outcome, we can infer that while the training accuracy is approximately 99.70%, the test accuracy without any data augmentation is approximately 70.3%.

**Question 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?**

**Data Augmentation**

This method can be used to increase the model's accuracy. Data augmentation is one method that enables reliable results to be obtained even with tiny datasets. To create new data, it involves introducing random modifications to the provided training examples. This technique increases the model's ability to successfully generalize by ensuring that it encounters a wide variety of images during training.   
All of the results below are based on the training sample of 1500 and the validation test of 500. displaying the enhanced images that were trained   
Test Precision: 77.6% The validation accuracy of 79% indicates better results than the preceding (Question 1), which could be due to the following factors:



Test Accuracy: 82.2% Better results than the previous (Question 1) are evident from the

validation accuracy of 82.20%, which may be attributed to the following factors:

The following factors have led to an improvement in the model's performance:

In addition, we added 500 (1000–1500) training samples, which helped us improve the featured extractions that resulted in better performance. This helped us raise test and validation accuracy by over 10%. Finally, we utilized data augmentation in addition to the convolution layer.

**Question 3: 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**

• Since we know that using more and more data will help to improve the model's performance, we are unable to identify the optimum sample size.   
• To do this, test sets comprising 500 samples and 2000 training samples were used.   
  
confirmation. I've discovered that test accuracy is higher with 1500 photos as opposed to training samples of 1000 and 2000 photos.   
• Training accuracy increases with 1000 training samples.   
• Increasing the training sample to 2000 while keeping the validation and test sets with 500 samples.

**Results :**

|  |  |  |  |
| --- | --- | --- | --- |
| **Training samples** | **Validation Accuracy** | **Test Accuracy** | **Data Augmentation** |
| 1000 | 72.8% | 70.03% | NO |
| 1500 | 82.2% | 82.2% | YES |
| 2000 | 83.4% | 80.4% | YES |

**Question 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 any and all optimization techniques to get best performance.**

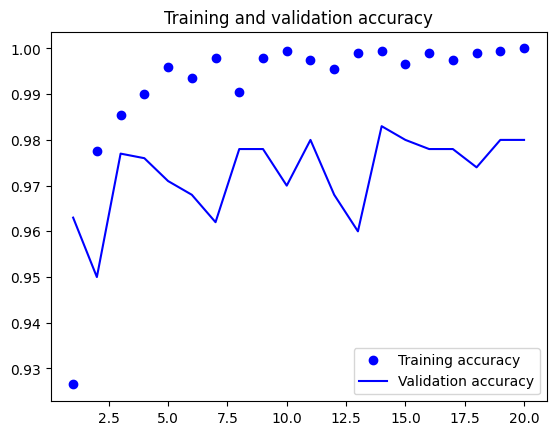
**Pre-trained model :**

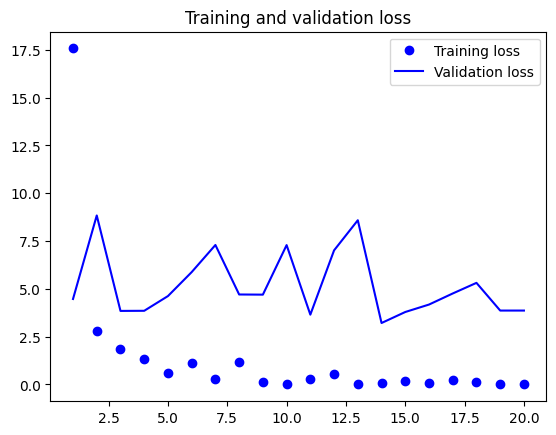
The main applications of trained networks are in feature extraction and fine-tuning.   
A pretrained network can be used as a generic model with its features applied to a range of computer vision applications if its starting dataset is large and diversified. The deep learning   
  
One of its main advantages over other machine learning techniques is its capacity to apply learnt attributes to a wide range of jobs. With 1.4 million annotated photos and 1,000 distinct classifications, the ImageNet dataset provides an example of studying a large trained convolutional neural network. The collection comprises many animal categories, such as various breeds of dogs and cats. This network uses the popular and straightforward VGG16 convolutional neural network architecture for ImageNet.

In this case, feature extraction will be used to improve the outcomes, initially without data augmentation and then with data augmentation.

**Pre-trained model with no augmentation :**

We have a 98% validation accuracy. The graphs show that, despite utilizing dropout at a rather high rate, we almost instantly reach overfitting.





**Pre-trained model utilizing augmented data:**

• 97.8% validation accuracy   
• 97.5% of trains are accurate.   
• 96.7% test accuracy.

**Fine-tuning a pretrained model**

• validation accuracy is 97.3%

• train accuracy is 99.6%

• test accuracy: 96.5%.

Perfecting a previously trained model

• 98% of validations are accurate.

The reduction of overfitting is the aim of data augmentation.   
Overfitting can be avoided in the following ways:

• Expanding the training sample isn't always an option. Data augmentation is one method of maximizing the limited amount of training data available.   
• How much overfitting happens as the model's size is reduced depends on the number of learnable parameters in the model, or the number of layers and units in layers.

**TABLE FOR PRE-TRAINED MODEL**

|  |  |  |
| --- | --- | --- |
| **Data Augmentation** | **Train Accuracy (%)** | **Validation Accuracy (%)** |
| NO | **100** | **98** |
| YES | **97.5** | **97.8** |

**CONCLUSION:**

We used a brief training set of 1000 samples and achieved a training accuracy of 92%.  
The reduction of overfitting is the aim of data augmentation.   
  
Techniques to avoid overfitting:   
  
• Expanding the training set isn't always a viable solution. Data augmentation is one method of maximizing the limited amount of training data available.   
• How much overfitting happens as the model's size is reduced depends on the number of learnable parameters in the model, or the number of layers and units in layers.   
• Zeroing off part of the layer's weights during training is a useful approach to reduce overfitting during training; this helps to regularize the weight values' distribution, avoiding or minimizing overfitting and lowering the complexity of a network.

• Eliminating part of the layer's output properties during training is a smart method to reduce overfitting. The percentage of traits that are null is referred to as the "dropout rate".

The sample sizes and model settings for the train, test, and validation sets are shown in the tables above. We offer the results for the original model without data augmentation as well as the trained models with augmentation using various train and validation sizes with a bigger train size. We compare the validation accuracy, accuracy, and data augmentation for the pre-trained model.   
Both a larger training set and a different size validation set increase the model's accuracy.

We compared the pre-trained model with and without data augmentation, and found that neither the model's accuracy nor its validation accuracy improved with data augmentation. Pre-trained models outperform custom models overall when there is a shortage of training data.