Image Processing with Deep Neural Network

Yash Amethiya, Aastha Patel, Rahil Bhensdadia, Priyavardhan Panchal

July 2020

1 Introduction

Deep learning have led to some significant breakthroughs in the last few decades. Integrating image processing with the deep learning, more specifically Convolutional Neural Network (CNN), simulates the visual cortex of human brain to analyze the visual feed of our surrounding.

Combining image recognition with neural networks forms a core component in various applications in e-commerce, gaming, home-automation, education etc. It can be used to perform tasks like identifying objects, labelling them, guiding autonomous vehicles and other self-assisting systems. To accomplish these tasks, image recognition algorithms are helpful, like image classifier, which takes an image as an input and predicts what it contains.

Based on that, a CNN(Convolutional Neural Network) model has been developed which follows similar classification algorithms which can identify a dog or a cat in a given picture. To implement such classification algorithms, first of all the model has to be trained with a pre-defined set of images, called a dataset, through which the model can learn how a dog or a cat looks like.

Before feeding these datasets, the images need to loaded and pre-processed with use of python libraries like TensorFlow and Keras. The accuracy of the image recognition will rely on the quality of the training and the testing datasets that are used to train the model.

2 Pre-processing Datasets

Some of the parameters taken into consideration before supplying data into the model:

- Image-quality : Images with higher quality gives more information to the model but also require more computing power to process
- Number of images : more the data we feed into the model, model will learn efficiently and will result in more accuracy.

- Image scaling : All the images have the same aspect ratio and size.Also images used for train the model are squared for maintaining uniformity.

- Number of channels : Since we are dealing with colored images of cats and dogs, there are 3 color channels (Red, Green, Blue), with colors represented in range [0,255].

3 Selecting the Layers and Training the Model

Selecting the Layers and its parameters needs many aspects to consider. The first thing to consider for CNN is the convolution layer.The convolution layer has these main component's to be considered,

- Number of Layers : In the current model, 5 layers are used. The initial layers will identify less detailed features. Gradually, the deeper layers distinguish distinct features from images provided in training set and can differ a cat from a dog.

- Number of filters : Here the number of filters increase as the layers proceed closer to the output and they increase in the power of 2 as it help to converge faster and hence the model learns faster.

- The size of kernel : The size of the kernel is chosen based on the size of the input images. As the images in input are squared of size 8X8, so kernel size is chosen as 3X3. Also kernel size must be an odd integer as it is symmetric about the origin.

- Padding: the need of padding arises if we want to prevent the reduction of dimension of Image while convolution process by padding extra zeroes at the sides and expanding the size of matrix.

- Strides :The length of each step while convolution along the X and Y axis is specified as strides.Normally strides are kept as (1,1) but increasing the strides can decrease the computations too. Its again the trade off between computation time and accuracy.

- Activation Function : The activation function is applied after the convolution process over the data.This function helps in keeping the values in a particular range which in turn helps us decrease the deviation among the value and increase the algorithms performance. Generally used activation functions are Re-Lu, sigmoid, step ,etc .We have used Re-Lu function for internal layers and soft-max for last layer .

- Batch Normalization :Batch normalization is a technique for training very deep neural networks that standardizes the inputs to a layer for each mini-batch. Here our model has batch of [batch-size] images.This helps in decreasing the epoch and stabilizing the learning process.

- Max Pooling : The layer of Max Pooling simply extracts the important features from the layer by selecting maximum value of pixel among a region and creating output where the region is replaced by that maximum value .Max pool also helps in reducing the size of data while keeping the important feature's significance

- Drop out : In this technique the neurons of current layer are randomly disconnected from the neurons of the next layer.This helps the model to generalize and prevent over-fitting.

The model composed in our application for classifying cats and dogs has following specifications:

- Structure: There are 4 bundle layers for convolution pass followed by 1 layer for flattening the data, 2 layers of dense(fully connected neural Net.) for generating output in accordance to classification .

- A convolution bundle: Each Bundle has a convolution layer, a batch Normalization layer, a max pool layer and a dropout layer

- Batch size: The batch size set for training the model is 15

- Number of Epoch : 25 are used for training

- Python packages used:

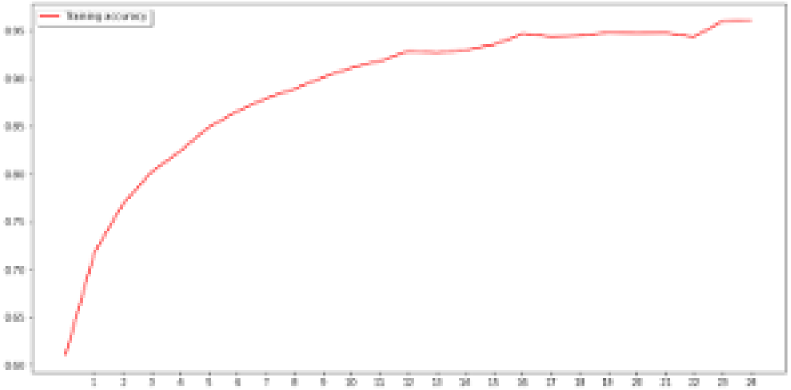
Packages
tensorflow
tensorflow.keras
sklearn
pandas
matplotlib
keras

Table 1: Python Packages

4 Results

With training for 4000 images of cat and dog and 2000 testing images the model obtained following results:

- Training statistics : final epoch loss=0.1019 ; accuracy=0.9606



Training Accuracy

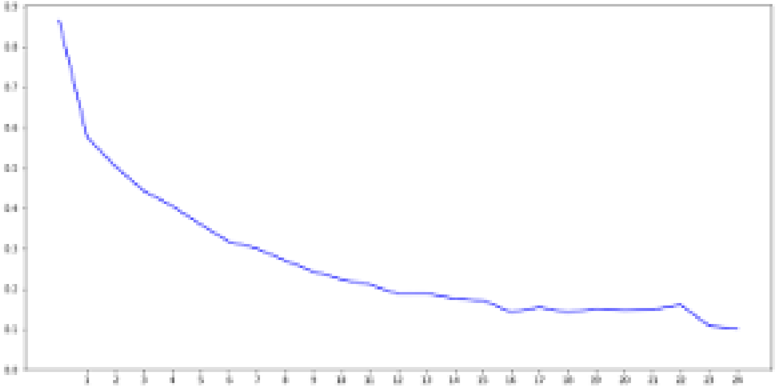
- Testing statistics:

-	Predicted Cat	Predicted Dog
is Cat	923	77
is Dog	133	867

Table 2: Confusion Matrix

type	precision	recall	f1-score	support
0	0.87	0.92	0.90	1000
1	0.92	0.87	0.89	1000
accuracy	-	-	0.90	2000
macro avg	0.90	0.90	0.89	2000
weighted avg	0.90	0.90	0.89	2000

Table 3: Classification Report



Training Loss

5 Conclusion

From this Model we have achieved the goal of classifying Cats and Dogs with 1780 correct predictions from 2000 test images. Which gives us an accuracy of 89.5