Parth Dave

18012011013

CE-A

Sem-7th

Batch-AB5

Project Title: Cats vs Dogs

1. Introduction

In this Keras project, we will build and train convolutional neural network for classifying images of Cats and Dogs. The input image will be analysed and then the output is predicted. The model that is implemented can be extended to a website or any mobile device as per the need. The Dogs vs Cats dataset used in this project is available on Kaggle.

2. Libraries:

Libraries used are numpy, pandas, keras, sklearn, maplotlib

a. Numpy:

NumPy offers comprehensive mathematical functions, random number generators, linear algebra routines, Fourier transforms, and more.

b. Pandas:

pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language.

c. Keras:

Keras is an API designed for human beings, not machines. **Keras** follows best practices for reducing cognitive load.

d. Sklearn:

Sklearn has Simple and efficient tools for predictive data analysis, accessible to everybody, and reusable in various contexts built on NumPy, SciPy, and matplotlib**.**

e. Maplotlib:

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.

3. Model Architecture:

Sequential model is used with three convolutional layers and two dense layers. Each convolutional layer has batch normalization, 2x2 max pooling and 0.25 dropout. The activation function used in convolutional layers is relu. The Dense layer has 512 nodes and relu activation. It also has batch normalization and 0.5 dropout. The output layer has two output nodes and softmax activation function. The model uses categorical crossentropy for loss and resprop optimizer.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 126, 126, 32)      896

batch_normalization (BatchN  (None, 126, 126, 32)     128
ormalization)

max_pooling2d (MaxPooling2D  (None, 63, 63, 32)       0
)

dropout (Dropout)            (None, 63, 63, 32)        0

conv2d_1 (Conv2D)            (None, 61, 61, 64)        18496

batch_normalization_1 (Batc  (None, 61, 61, 64)       256
hNormalization)

max_pooling2d_1 (MaxPooling  (None, 30, 30, 64)       0
2D)

dropout_1 (Dropout)          (None, 30, 30, 64)        0

conv2d_2 (Conv2D)            (None, 28, 28, 128)       73856

batch_normalization_2 (Batc  (None, 28, 28, 128)      512
hNormalization)

max_pooling2d_2 (MaxPooling  (None, 14, 14, 128)      0
2D)

dropout_2 (Dropout)          (None, 14, 14, 128)       0

flatten (Flatten)            (None, 25088)             0

dense (Dense)                (None, 512)               12845568

batch_normalization_3 (Batc  (None, 512)              2048
hNormalization)

dropout_3 (Dropout)          (None, 512)               0

dense_1 (Dense)              (None, 2)                 1026

=================================================================
Total params: 12,942,786
Trainable params: 12,941,314
Non-trainable params: 1,472
_________________________________________________________________
```

Training accuracy is 83.64 and validation accuracy is 85.21

4. Dataset:

Name of the dataset is cats vs dogs and The dataset includes 25,000 images with equal numbers of labels for cats and dogs. Asirra(Animal Species Image Recognition for Restricting Access) is a HIP that works by asking users to identify photographs of cats and dogs. This task is difficult for computers, but studies have shown that people can accomplish it quickly and accurately. This dataset link is available on: https://www.kaggle.com/c/dogs-vs-cats/data

5. Output:

```
sample_test = test_df.sample(9)
sample_test.head()
plt.figure(figsize=(12, 24))
i=1
for index, row in sample_test.iterrows():
    filename = row['filename']
    category = row['category']
    img = load_img("/content/drive/MyDrive/dl_project/catdog/test1/"+filename, target_size=Image_Size)
    plt.subplot(6, 3, i)
    i += 1
    plt.imshow(img)
    plt.xlabel("{}".format(category.upper()))
plt.tight_layout()
plt.show()
```

This code is available on my github:

https://github.com/parthdave11/Cats-vs-Dogs