Project 6 Report

ENPM 673 - Perception for Autonomous Robots

Team 5

Prof. Dr. Mohammed Charifa

UNIVERSITY OF MARYLAND

Nikhil Lal Kolangara (116830768)
Kartik Venkat (116830751)
Kushagra Agrawal(116700191)

This paper represents our own work in accordance with University regulations.

Contents

1 Introduction

In deep learning, a convolutional neural network (CNN) is a class of deep neural networks, most commonly applied for analyzing visual imagery. A convolutional neural network consists of an input and an output layer, as well as multiple hidden layers. The hidden layers of a CNN typically consist of a series of convolutional layers that convolve with a multiplication or other dot product. The activation function is commonly a RELU layer, and is subsequently followed by additional convolutions such as pooling layers, fully connected layers and normalization layers, referred to as hidden layers because their inputs and outputs are masked by the activation function and final convolution.

The aim of this project is to classify whether images in the dataset contains cat or dog.

2 Pipeline

Step 1: Data Preparation

The dataset was provided from Kaggle and has already been segregated into train and test folder. These folders consists of the images of various cats and dogs.

These photos are color images and are of different sizes and shapes. Therefore in order for modeling, it is necessary for these images to be reshaped. The technique used to reshape these images are done by using the resize function to make all images of 96*96 square size.

The coloured image is converted to numpy array.The dataset contains the image of cats and dogs, the name of the image i.e. cat or dog is extracted. The input images are standardized to the range 0 to 1. Scikit learn multilabel binarizer is used to binarize the cat and dog labels.

Step 2: Training

The preprocessed dataset is split into training and test (80 percent data is used for training) and we split the labels also into training and test. Scikit learn module is used to split the preprocessed dataset into training and testing dataset.

Data image generator is used to create the augmented data. Then the augmented data is sent to the below architecture.

Step 3: Architecture

The architecture that is being used for training is smaller VGGNet.

VGGNet-like architectures are characterized by:

1. Using only $3 * 3$ convolutional layers stacked on top of each other in increasing depth.

2. Reducing volume size by max pooling.

3. Fully-connected layers at the end of the network prior to a softmax classifier.

The first convolution layer has 32 filters with a 3x3 kernel. The activation function used is RELU followed by batch normalization. The max POOLING layer uses a 3 x 3 POOL size to reduce spatial

dimensions quickly from 96 x 96 to 32 x 32. Dropout of 0.25 is also utilized in the network architecture. Dropout works by randomly disconnecting nodes from the current layer to the next layer. This process of random disconnecting during training batches helps naturally introduce redundancy into the model — no one single node in the layer is responsible for predicting a certain class, object, edge, or corner.

In the next layer multiple convolutional layers have been stacked together(prior to reducing the spatial dimensions of the volume). This allows us to learn a richer set of features. The filter size has been increased from 32 to 64 . The deeper we go in the network, the smaller the spatial dimensions of the volume, and the more filters the model learn. The max pooling size is reduced from 3 x 3 to 2 x 2 to ensure that the spatial dimensions does not reduce too quickly. Dropout is again performed at this stage.

Again two convolutional layers have been stacked together but the filter size has been increased to 128. Dropout of 25 percent of the nodes is performed to reduce overfitting again.

The final layer which is the fully connected layer, is specified by Dense(1024) with a rectified linear unit activation and batch normalization. Dropout of 50 percent is performed a final time. In the final step the model is round out with a softmax classifier.

Step 4: Model Compilation

The above generated model is compiled using binary cross entropy function and Adam optimizer.

The data is then fit into this model where the inputs are augmented training data, validation data and epochs.

The batch size used in our project is 32 and the data is trained for 20 epochs. The model is then saved which is used to then classify our images that are inputted.

Step 5: Testing

The testing images are sorted and are preprocessed according to the same procedure used for the training dataset.

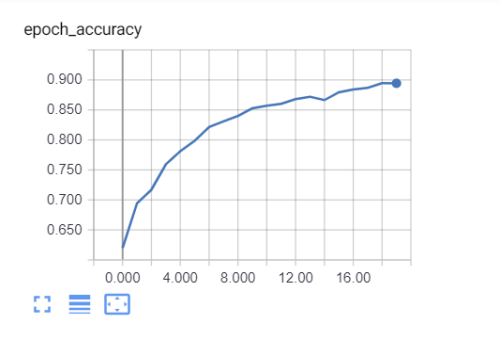
The trained model is imported and the testing images are sent to the trained model, the prediction is done whether the image is of cat or dog.

The prediction is being written in the .csv file and the classification is done.

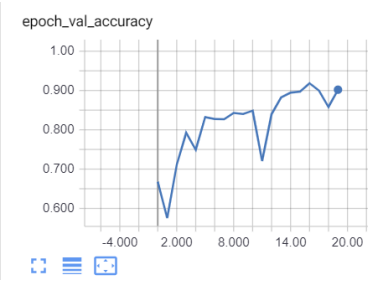
3 Output

Data could have been trained for more epochs but 20 epochs gave the desired results.

Learning rate is 0.001. Initially learning rate of 0.01 was used but then the number of epochs had to be increased thus increasing the learning time.

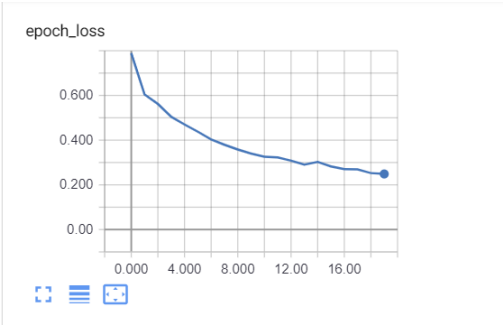


(a) Training Acuuracy

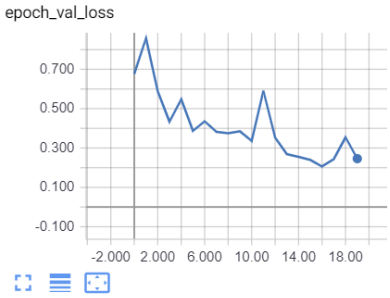


(b) Testing Accuracy

Figure 1: Accuracy Graph



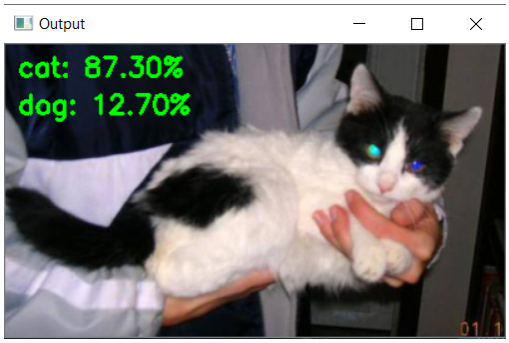
(a) Training Loss



(b) Testing Loss

Figure 2: Loss Graph

(a) Result on Dog's Image



(b) Result on Cat's Image

Figure 3: Results Obtained

SN	Image Name	Dog (% prediction)	Cat (% prediction)	Binary Classification
1	1	97.56085277	2.439141646	1
2	2	99.89031553	0.109678286	1
3	3	99.99734163	0.002659085	1
4	4	99.96389151	0.036102373	1
5	5	14.12195265	85.87804437	0
6	6	6.715676934	93.28432679	0
7	7	0.197494309	99.80250597	0
8	8	31.33172393	68.66827011	0
9	9	3.761742264	96.23826146	0
10	10	0.457244646	99.54274893	0
11	11	0.479969988	99.52002764	0
12	12	99.96880293	0.031202062	1
13	13	20.96813619	79.03186679	0
14	14	12.69568205	87.30432391	0
15	15	1.415268518	98.58472943	0
16	16	3.182668984	96.81733251	0
17	17	78.90226841	21.09773904	1
18	18	99.98408556	0.015916211	1
19	19	6.114456803	93.88554692	0
20	20	7.227522135	92.77247787	0

Figure 4: CSV data containing the first 20 Images

4 References

1. https://www.youtube.com/watch?v=WvoLTXIjBYU&list=PLQVvvaa0QuDfhTox0AjmQ6tvTgMBZBEXN&index=3

2. https://www.youtube.com/watch?v=HMcx-zY8JSg&list=PL9Hr9sNUjfsmEu1ZniY0XpHSzl5uihcXZ&index=4

3. https://www.pyimagesearch.com/2018/04/16/keras-and-convolutional-neural-networks-cnns/