Name:- Swapna Vaidya ***Class:-*** MSDS – 422

COLAB NOTEBOOK LINK :- https://colab.research.google.com/drive/14gatMCneuhYcBZN0soS7g19PLdUVRli1

Convolutional Neural Networks Models

(1) ***A summary and problem definition for management***

This is a Kaggle competition data set for Dogs V/S Cats Redux: Kernel Edition. In the training dataset there are 25000 images of dogs and cats. While the test dataset has 12,500 images.

Using this dataset, we need to advice a website provider who is looking for tools to automatically label images provided by end users. As we look across the factors in the study, making recommendations to management about image classification, we are most concerned about achieving the highest possible accuracy in image classification. That is, we should be willing to sacrifice training time for model accuracy. What type of machine learning model works best? If it is a convolutional neural network, what type of network should we use? Part of this recommendation may concern information about the initial images themselves (input data for the classification task). What types of images work best?

(2) ***Discussion of the research design, measurement and statistical methods, traditional and machine learning methods employed***

Based on the exploratory data analysis that was done the survey data, below were some of the findings:-

- There are 2 sets of data provided:- Training dataset and Test dataset.
- In the training data set there are 25000 images with the file name prefixed as cats or dogs
- In the test data set there are 12000 images with an id prefixed to the file name

(3) ***Overview of programming work***

This assignment was done in Google Colab, using 4 main libraries (Pandas, numpy, Skylearn and Keras).

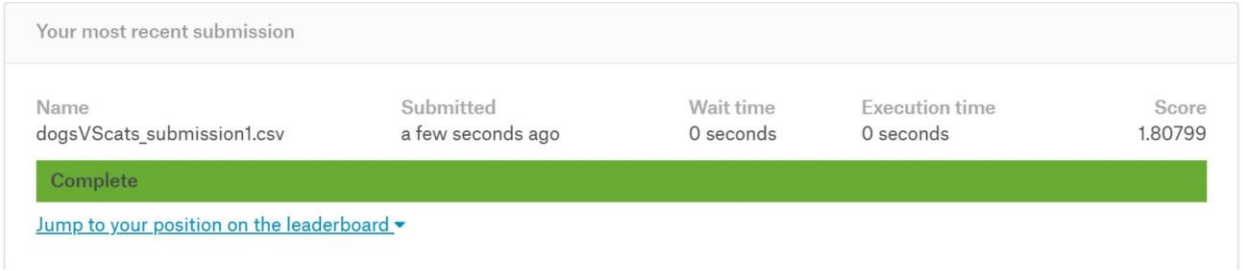
- ***Ingestion:-*** Using Google drive internal tools the zip files were unzipped into training and test data set. All the images under the train directory were stored into train_images_dogs_cats array with the entire filepath. The same is done for the test folder too.
- ***Data Cleaning and Transformation:-*** Based on the name of the image which is an id, sort the images in an ascending order. Do the same for the test data.
- ***Modeling:-*** In the training dataset, since all the images of dogs and cat are in the training directory with no labels, the data needs to be parsed in such a way that the labels can be stored separately. Hence a custom function is written where the image file name is parsed to get the word “dog” or “cat” and store the value as 1 for dog and 0 for cat. At the same time the images are read by reshaping using cv2.resize method. Next the training data is then split into train and validation dataset of 80:20 ratio. Similarly using ImageDataGenerator function the parameters are set to further rescale and standardize using rescale, shear_range, zoom-range and horizontal_flip methods. Next the data using the flow method is finally preprocessed with the parameters set in ImageDataGenerator function.

  The base model used was a Squential. 3 convolutional layers were used with 32,32 and 64 filters. In addition, the activation method used was “Relu”. Finally, the MaxPooling2D layer was also used. The final layer before the output was flattened with a dropout of 0.5.

  The model was then compiled with a binary_crossentropy loss function, with an ‘RMSprop’ optimizer and metrics as accuracy. The model was then fit with the fit_generator function using 30 epochs.

  Similar iteration of the model were tried with different filters for the convolution layer and different epochs.

  Similar steps of preprocessing, standardizing and scalarizing were done on the test data. Using the test data, the predict method was used on it to find the probabilities. Finally, the output file was generated with the label which is probabilities and Id which for the images. This output file was then submitted to Kaggle for scoring. The best Kaggle score that I got was for my first model:-



Review of results with recommendations for management

Based on the above modeling and log loss score my recommendation to the website provider would be the following:-

- To go with Convolution Neural Network instead of other machine learning techniques to label the images provided by end users.
- While using CNN networks, definitely use runtime as GPU instead of CPU or even TPU. With CPU depending on the size of the training dataset, it either crashes or runs very slowly. With GPU it atleast runs the model and is faster than CPU.
- Keep the CCN model simple, using 3 convolution layers with 32 to 64 filters and dropout of 0.5.
- Before the data is fed into the model ensure that it is preprocessed, standardized or scalarized.
- Keep the epochs to 30. Increasing the epochs increases the fit time but not much improvement in the accuracy.
- Similarly when training the model, instead of using the entire dataset, you can use 1/3 of the data to train the model, hence reducing the processing time but still getting sufficient accuracy!