Machine Learning Engineer Nanodegree

Capstone Proposal

Deepthi
July 11, 2019

Proposal

Domain Background:

To assess the impact of climate change on Earth's flora and fauna, it is vital to quantify how human activities such as logging, mining, and agriculture are impacting our protected natural areas.

Researchers in Mexico have created the VIGIA project, which aims to build a system for autonomous surveillance of protected areas. A first step in such an effort is the ability to recognize the vegetation inside the protected areas. In this project, we are tasked with the creation of an algorithm that can identify a specific type of cactus called columnar cactus (Neobuxbaumia tetetzo) in aerial imagery.

Data:

The dataset contains a large number of 32 x 32 thumbnail images containing aerial photos of a columnar cactus (Neobuxbaumia tetetzo). Kaggle has resized the images from the original dataset to make them uniform in size. The file name of an image corresponds to its id. Data consists of 17500 training files and 4000 test files.

Image samples from train set:

From the benchmark model we can clearly see that there is imbalance in the data. We have a biased dataset. We have has_cactus for more data equal to 1.

Files:
train/ - the training set images
test/ - the test set images (labels to be predicted)
train.csv - the training set labels, indicates whether the image has a cactus (has_cactus = 1)
sample_submission.csv - a sample submission file in the correct format.

Kaggle Competition Link:
https://www.kaggle.com/c/aerial-cactus-identification

Kaggle Datasets Link:
https://www.kaggle.com/c/aerial-cactus-identification/data

Solution Statement:

A deep learning algorithm will be developed using Tensorflow/Keras and will be trained with training data. Specifically a CNN will be implemented in Tensorflow/Keras and will be optimized to minimize area under the ROC curve between the predicted probability and the observed target as defined in the Evaluation Metrics section. Predictions will be made on the test data set and will be evaluated.

Benchmark Model

The model with the Public Leaderboard AUC score of 0.9703 will be used as a benchmark model. Attempt will be made so that score (Area under the ROC curve) AUC obtained will be among the top 50% of the Public Leaderboard submissions.

The architecture of the benchmark model is explained below:
Keras with Data Augmentation-

- Connected layers of Conv2D for extracting features from a small 3x3 kernel.
- The number of filters has been increased as the model goes deep to extract more features.
- MaxPooling and Dropout used.
- Densely connected layer.
- The activation function is ReLU in all Convolutional layers.
- Since we want to predict the probabilities last layer used is SoftMax Layer.
- We have a binary classification problem hence the loss function used is binary_crossentropy.

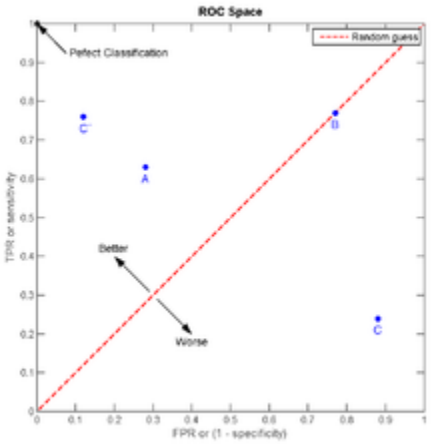
- Adam optimizer and accuracy metrics have been used.

Evaluation Metrics

Submissions are evaluated on area under the ROC curve or AUC between the predicted probability and the observed target.

What is ROC Curve?

A receiver operating characteristic curve, or ROC curve, is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied.



Source: Wikipedia

Submission File

For each ID in the test set, we must predict a probability for the has_cactus variable. The file should contain a header and have the following format:
id,has_cactus
000940378805c44108d287872b2f04ce.jpg,0.5
0017242f54ececa4512b4d7937d1e21e.jpg,0.5
001ee6d8564003107853118ab87df407.jpg,0.5
etc.

Project Design

From the description and problem statement it can be inferred that computer vision can be used to arrive at a solution. CNN class of deep learning algorithm can be employed

for this problem. Initially data exploration will be carried out to understand possible labels, range of values for the image data and order of labels.

Data should be processed into appropriately pre-processed floating-point tensors before being fed to our network. So, the steps for getting it into our network are roughly

- Read the picture files
- Decode JPEG content to RGB pixels
- Convert this into floating tensors
- Rescale pixel values (between 0 to 255) to [0,1] interval.

We will make use of ImageDataGenerator method available in keras to do all the preprocessing. This will help preprocess the data and can end up with better predictions.

Now, after preprocessing is done with our data, we will split our dataset to training and validation for training our model and validating the result respectively and CNN will be implemented in Tensorflow/Keras. Finally, necessary predictions on the test data will be carried out and AUC will be evaluated between the predicted probability and the observed target.