Machine Learning Engineer Nanodegree

Capstone Proposal

Firdos Rehman June 1st, 2018

Proposal

Domain Background

This project is to classify the different leaves. Out of approximately half a million species of the plants in the world, it is very problematic to classify the different species and it had been seen historically that many of them results in duplicate identifications. We can apply automating the plant recognition so that the it helps it helps in many applications like in preserving and species population tracking, Plant - based medicinal research which many people are choosing as it is safer and lesser side effects, ecological reasons, crop and food supply management

The objective is to use binary leaf images and extracted features, including shape, margin & texture, to accurately identify 99 species of plants. Leaves, due to their volume, prevalence, and unique characteristics, are an effective means of differentiating plant species. They also provide a fun introduction to applying techniques that involve image-based features. As a first step, try building a classifier that uses the provided pre-extracted features. Next, try creating a set of your own features. Finally, examine the errors you're making and see what you can do to improve.

Kaggle hosted this competition for the data science community to use for fun and education. This dataset originates from leaf images collected by
James Cope, Thibaut Beghin, Paolo Remagnino, & Sarah Barman of the Royal Botanic Gardens, Kew, UK.
Charles Mallah, James Cope, James Orwell. Plant Leaf Classification Using Probabilistic Integration of Shape, Texture and Margin Features. Signal Processing, Pattern Recognition and Applications, in press. 2013.

Originally the dataset was hosted by UCI machine learning repository.

Problem Statement

This project is to build a model which provides highest accuracy to find the species of plants from characteristics of the leaves.

The objective is to use binary leaf images and extracted features, including shape, margin & texture, to accurately identify 99 species of plants. Leaves, due to their volume, prevalence, and unique characteristics, are an effective means of differentiating plant species. They also provide a fun introduction to applying techniques that involve image-based features. As a first step, try building a classifier that uses the provided pre-extracted features. Next, try creating a set of your own features. Finally, examine the errors you're making and see what you can do to improve.

Datasets and Inputs

The dataset consists approximately 1,584 images of leaf specimens (16 samples each of 99 species) which have been converted to binary black leaves against white backgrounds. Three sets of features are also provided per image: a shape contiguous descriptor, an interior texture histogram, and a fine-scale margin histogram. For each feature, a 64-attribute vector is given per leaf sample.

Note that of the original 100 species, we have eliminated one on account of incomplete associated data in the original dataset.

File descriptions

- train.csv - the training set
- test.csv - the test set
- sample_submission.csv - a sample submission file in the correct format
- images - the image files (each image is named with its corresponding id)

Data fields

- id - an anonymous id unique to an image
- margin_1, margin_2, margin_3, ..., margin_64 - each of the 64 attribute vectors for the margin feature
- shape_1, shape_2, shape_3, ..., shape_64 - each of the 64 attribute vectors for the shape feature
- texture_1, texture_2, texture_3, ..., texture_64 - each of the 64 attribute vectors for the texture feature

Solution Statement

The dataset that I am using is small and has more features. Roughly 300 features were provided. I would extract approximately 64 features from the images, design a convolutional network and do the job of classifying and using that network to pick the bottle neck features and append the bottle neck features to the columns of features that were provided and then run other models on it. I will be using feature engineering but using Convolutional network to create the features. We have more features and it may lead to overfitting. Using Neural networks we can have dropouts and less features. We have to make sure it doesn't overfit.

I would like to use multilayer perceptron and the Neural networks in addition to the benchmark model.

Benchmark Model

Equal Probability Model which shows that there is an equal probability of being from each plant or any species. This will be the baseline benchmark model. Then I will use Logistic Regression and Random Forest models. I will use 2 classifiers and then see which gives me the best accuracy and the best best fit

https://www.kaggle.com/jeffd23/10-classifier-showdown-in-scikit-learn (https://www.kaggle.com/jeffd23/10-classifier-showdown-in-scikit-learn)

Evaluation Metrics

Submissions are evaluated using the multi-class logarithmic loss. Each image has been labeled with one true species. For each image, you must submit a set of predicted probabilities (one for every species). The formula is then, logloss=$-1/N\sum_{i=1}^{N}\sum_{j=1}^{M} yij \log(Pij)$

where N is the number of images in the test set, M is the number of species labels, log is the natural logarithm, yij is 1 if observation i is in class j and 0 otherwise, and p ij is the predicted probability that observation i belongs to class j.

The submitted probabilities for a given device are not required to sum to one because they are rescaled prior to being scored (each row is divided by the row sum), but they need to be in the range of [0, 1]. In order to avoid the extremes of the log function, predicted probabilities are replaced with

$$max(min(p, 1-10^{-15}), 10^{-15})$$

Project Design

- Data analysis
- Feature extraction
- Model Building - Logistic Regression, Random Forest, Convolutional Neural networks that gives highest accuracy. Exploring the application of deep learning with different number of nodes, hidden layers and activation functions.
- Performance evaluation

References:

Kaggle Competition: https://www.kaggle.com/c/leaf-classification (https://www.kaggle.com/c/leaf-classification)

Kaggle Kernel: https://www.kaggle.com/jeffd23/10-classifier-showdown-in-scikit-learn (https://www.kaggle.com/jeffd23/10-classifier-showdown-in-scikit-learn)

http://lamda.nju.edu.cn/weixs/project/CNNTricks/CNNTricks.html (http://lamda.nju.edu.cn/weixs/project/CNNTricks/CNNTricks.html)