Udacity Machine Learning Nanodegree

Google Landmark Recognition Challenge

Steve Tsiolis

January 31st, 2019

Proposal

Domain Background

Understanding where the field of deep learning relates to machine learning an AI as a whole is commonly misunderstood these days. It is no secret that AI has been taking off recently and is competitively used in world-wide industry. Many learning algorithms have been developed over the past years such as linear regression, K-Means, Decision trees, Random Forest, SVM and Artificial Neural Networks. Deep learning is making use of Artificial Neural Networks[1].

Deep learning has allowed for the design of object recognition and is becoming more powerful by the day. A good real-life example of this is when you go to take a picture of yourself (a "selfie") and notice that there is a square around your face and any other faces in the picture[2]. Similarly, deep learning with deep neural networks can be used in areas to identify disease in plants[3]. This is accomplished under the hood using layers upon layers of convolutional neural networks which are constantly outperforming the previously listed machine learning techniques on image recognition.

Problem Statement

A great obstacle to landmark recognition research is the lack of large annotated datasets. Thankfully Google has presented machine learning enthusiasts the largest worldwide dataset on Kaggle.com[4] boasting 15,000 classes. The main objective of this challenge is to build a model which recognizes the correct landmark (if any) in a dataset of challenging test images.

Datasets and Inputs

The Google-Landmark Recognition challenge provides two main sets of data, test.csv and train.csv. In the train.csv file, both the URL and landmark_id is provided and hence there is

[1] https://medium.com/intuitionmachine/why-deep-learning-is-radically-different-from-machine-learning-945a4a65da4d
[2] http://cis.csuohio.edu/~sschung/CIS660/DeepFaceRecognition_parkhi15.pdf
[3] https://www.hindawi.com/journals/cin/2016/3289801/abs/
[4] https://www.kaggle.com/c/landmark-recognition-challenge

one landmark park URL. In the test.csv file, a random photo is given which could include a random amount of landmarks per URL i.e no landmark, one landmark, or multiple landmarks. Images can be formatted differently, i.e different shape and size and include colour layers.

For the purpose of completing this challenge on a personal computer, I have restricted the size of the datasets as they are extremely large and it can not be guaranteed that all the URL's exist as this dataset is many months old. There are 1,048,575 images available for use, instead of using all 15,000 individual landmarks, I have decided to use the first 150 landmarks and take 3% of images from each ID in order to preserve the ratio of images which reduces the number of images to 11,606. These have been chosen in the order of most frequently occurring images in the dataset. The classes are not extremely well balanced but in the 150 class sample it is reasonably balanced.

Solution Statement

I will be making use of both the Keras and TensorFlow library to create a CNN model. The CNN model will likely contain convolutional layers to generate feature maps, pooling layers to reduce overfitting, flattening and fully connected layers to help extract the global relationship between features. Image will need to be resized and contain useless information so Data Augmentation[5], Dropout layers[6] and Batch Normalization layers[7] will be explored to increase stability. This will of course be decided as the project progresses in order to give the best accuracy of determination.

Benchmark Model and Evaluation Metrics

Considering there is a reasonably balanced dataset, the accuracy of the model can be given by using the number of landmark id's in the test dataset and the number of images in the test dataset. This is calculated as such:

$$\text{Accuracy} = \left(\frac{\sum_{x=0}^{x=N}\left(x \cdot \frac{n_{landmark_{test}}}{n_{images_{test}}}\right)}{n_{images_{test}}}\right) = \frac{Number\ of\ correctly\ predicted\ classes}{Total\ number\ of\ predictions}$$

A benchmark can be created by guessing the landmarks at random, it is expected that the model will give a higher accuracy than this.

[5] https://towardsdatascience.com/augmentation-for-image-classification-24ffcbc38833
[6] https://machinelearningmastery.com/dropout-regularization-deep-learning-models-keras/
[7] http://cs231n.github.io/convolutional-networks/

The official GAP (Global Average Precision) at k, where k=1 metric is used as the evaluation technique officially by Google and is as such:

$$GAP = \frac{1}{M}\sum_{i=1}^{N} P(i)rel(i)$$

Where:

- N is the total number of predictions returned by the system, across all queries
- M is the total number of queries with at least one landmark from the training set visible in it (note that some queries may not depict landmarks)
- P(i) is the precision at rank ii
- rel(i) denotes the relevance of prediciton ii: it's 1 if the ii-th prediction is correct, and 0 otherwise

This evaluation metric will also be looked into to see if results are consistant with accuracy as it is possible that accuracy measurements end up affected due to the skewness (poor balance) of the data.

Project Design

The Google-Landmark Recognition challenge as presented on the Kaggle website will be undertaken with the intent to identify landmarks with the highest accuracy possible from a dataset provided by Google. As the dataset contains 15,000 landscape images, a snippet of this will be used based on how frequently they appear.

Through the use of the Keras and TensorFlow libraries, a CNN model will be created potentially making use of convolutional layers, pooling layers, fully connected layers, dropout layers and batch normalization layers. The use of these layers are subject to the snippet of the dataset used and how it affects the accuracy.

A benchmark based on random guessing will be used to compare against the accuracy of the model.