Machine Learning Engineer Nanodegree

Capstone Proposal

Amr Elbeleidy
November 27, 2017

Proposal

Domain Background

Traditionally machine learning algorithms have been mainly used to solve classification problems. One machine learning domain that has gained traction and shown relatively high success rates is deep learning. Deep Learning is primarily concerned with finding representative models that have multiple levels of abstraction. Deep learning has been successfully used in a number of image classification applications where computers were able to achieve better than human performance.

Machine learning algorithms can be given labeled or unlabelled data to learn from. Labelled data is data that already contains the target answer we want to know. Unlabelled data is data that does not. When the machine learning algorithm is given labelled data, that is called supervised learning and when it is given unlabelled data, that is unsupervised learning.

In unsupervised learning, the computer is asked to detect or infer a pattern or structure in the data that we cannot readily see. One method of unsupervised learning is clustering. When clustering, the computer attempts to split the data to a number of sets where the data in each set are more similar to each other, in one or more aspects, than they are to data in other sets.

http://docs.aws.amazon.com/machine-learning/latest/dg/collecting-labeled-data.html

Problem Statement

We'd like to give to the computer a number of images and ask it to cluster the images that are more like each other together. We expect that if given images of n different entities, and asked to cluster them in n groups, the algorithm would group all images of the same entity together in the same cluster..

Datasets and Inputs

I propose to use the Oxford IIIT Pet Dataset, available at http://www.robots.ox.ac.uk/~vgg/data/pets/. The dataset has images of 37 different categories with approximately 200 images for each category. Each category represents a breed of either cats or dogs. This would allow us to investigate the ability of the algorithm to separate the different breeds as well as separate all cats and dogs.

Solution Statement

I propose to use a convolutional network to produce a meaningful representation of the images, that would then be fed to a clustering algorithm for separation. In order for the convolutional network to generalise the images, I intend to use pre-trained networks such as VGG19 with weights based on ImageNet classification. Since these are world leading networks in classification accuracy, it is possible to assume that they are able to detect patterns in images well.

The top fully connected layer of these classifiers would be removed so that the clustering algorithm can have direct access to the representations built by the convolutional network. Different clustering algorithms can then be compared in order to arrive at the best model.

Benchmark Model

Ideally we would produce a benchmark model by asking human subjects to separate the images themselves and then measure their performance. We would then compare the algorithm's performance to human performance.

Since I do not have the resources to set up such a benchmark model, and we are using a dataset that has labels available. We can use perfect clustering of images as our benchmark, that is if all the images with the same labels in the dataset end up in the same cluster by the algorithm.

Evaluation Metrics

Although clustering is an exercise of unsupervised learning, unlike most problems of this type, we are actually using a labelled dataset here. We do not feed the labels to the algorithm, and we do not ask the algorithm to give a meaningful label to the data. The only value in the cluster number the algorithm assigns to a particular image comes from whether it is the same, or different to, the cluster number of other images of the same entity.

In the simple case of asking the algorithm to separate between images of, say, cats and dogs. We would manually look at the results and assign the cluster number with a majority of cats to

the cat label, and the cluster number with the majority of dogs to the dog label. We can then proceed to use typical classification problem evaluation metrics, for example, a confusion matrix, accuracy and F1 score. In the case where the number of clusters is more than two, this becomes a multilabel classification problem and we can then treat each pair of classes as a separate classification problem and average out the f1 scores across the problems using different weighting options.

http://scikit-learn.org/stable/modules/model_evaluation.html

Project Design

A python environment will be set up, using Keras with a tensorflow backend to model the convolutional network. The scikit-learn library will be used for clustering and evaluation

I will separate the images for each problem in different directories, with images of each label placed in different subdirectories for easy identification. The OpenCV library will be used to read and resize the images into python or NumPy arrays of equal size as needed to prepare them for processing.