Machine Learning Engineer Nanodegree

Capstone Proposal

Prateek Gupta
June 19th, 2018

➢ Domain Background

In 1995's Clueless, you may recall Cher Horowitz using cutting-edge software to select her plaid ensemble. Cher's machine could identify chic head-to-toe looks, adding a small dose of sci-fi to the rom-com classic. Twenty-two years later, the 90's fiction movie is closer than ever to reality: Artificial intelligence in fashion is here.

Online fashion will be transformed by a tool that understands taste. Because if you understand taste, you can delight people with relevant content and a meaningful experience. Outfits are the asset that would allow taste to be understood, to learn what people wear or have in their closet, and what style each of us like.

➢ Problem Statement

Customers can purchase wide variety of fashion products like T-shirts from an e-commerce platform. One of the major fashion attributes of a T-shirt is the type of graphic present on it. Graphic type plays a major role in identifying T-shirt categories and determining its saleability. Some example graphic types are:

Biker Varsity Music

The challenge is to identify the graphic type of a T-shirt. Take in an image as an input and give graphic type of the T-shirt as the output. The model can be used during cataloguing new T-shirts on any e-commerce platform. With graphic type present as an attribute, customers will be able to search and filter for different type of T-shirts.

➢ Datasets and Inputs

The dataset is provided as the files **train.csv** and **test.csv**, with the information necessary to make a prediction. The dataset was labelled by identifying the graphic type for each T-shirt captured from various angles.

The format of each of the file is as follows:
<brand_name>, <article_type>, <gender>, <color>, <image_path>, <graphic_type>

Each image correlated with a link is in the *jpeg* format. The resolution of each image is 1080 x 1440, with the DPI of 72 pixels/inch.

The dataset features 24 different classes of graphic types for T-shirts. Training set (**train.csv**) includes 30000 rows (labelled images) and the testing set (**test.csv**) includes 70000 rows. Images are guaranteed to be of fixed dimensions.

I plan to scale down the image sizes, likely down to 25% of the original size. Since, the data is already *labelled*, it will save the time as required for coming up with the same using the CV techniques of undistorting and perspective transform.

➢ Solution Statement

As deep learning techniques have been very effective in image classification over the years, in this project, transfer learning along with data augmentation will be used to train a convolutional neural network to classify images of T-shirts to their respective classes. Transfer learning refers to the process of using the weights from pre-trained networks on large dataset. Fortunately many such networks such as RESNET, Inception-V3, VGG-16 pre-trained on *imagenet challenge* is available for use publicly. Specifically, I plan to use a convolutional neural network (CNN), which are very effective at finding patterns within images by using filters to find specific pixel groupings that are important.

➢ Benchmark Model

Random choice: We predict equal probability for a T-shirt to belong to any class of the 24 classes for the naive benchmark.

A well-designed convolutional neural network should be able to beat the random choice baseline model easily. However, due to computational costs, it may not be possible to run the transfer learning model with VGG-16 architecture for sufficient number of epochs so that it may be able to converge.

➢ Evaluation Metrics

The metric used for this project is multi-class logarithmic loss (also known as **categorical cross entropy**). Here each image has been labelled with one true class and for each image a set of predicted probabilities should be submitted. **N** is the number of images in the test set, **M** is the number of image class labels, $\mathbf{y_{ij}}$ is 1 if observation **i** belongs to class **j** and **0** otherwise, and $\mathbf{p_{ij}}$ is the predicted probability that observation **i** belongs to class **j**. A perfect classifier will have the log-loss of 0.

Multiclass log-loss punishes the classifiers which are confident about an incorrect prediction. If the class label is 1 (the instance is from that class) and the predicted probability is near to 1 (classifier predictions are correct), then the loss is really low, so this instance contributes a small amount of loss to the total loss and if this occurs for every single instance (the classifiers is accurate) then the total loss will also approach 0.
On the other hand, if the class label is 1 (the instance is from that class) and the predicted probability is close to 0 (the classifier is confident in its mistake), as *log(0)* is undefined it approaches infinity, so theoretically the loss can approach infinity.

➢ Project Design

- **Programming language:** Python 2.7+
- **Libraries:** Keras, Tensorflow, OpenCV, NumPy, Pandas
- **Workflow:**
  - Establishing the baselines with random choice for comparison.
  - Training a small convolutional neural network from scratch for further comparison with transfer learning models.
  - Extracting features from the images with the pre-trained network and running a small fully connected network 24 output neurons on the last layer to get predictions.
  - Fine tuning the pre-trained network by choosing different optimizers and by training the network on this dataset from the convolutional layers instead of the dense layers as long as it's computationally inexpensive.
  - Optionally, comparing the performance of multiple pre-trained networks. However, as fine-tuning them is computationally expensive, different pre-trained networks can be compared at the feature extraction stage instead of direct comparison.