Genre Classification of YouTube Videos

Modzelewski, Derek
dmodzel2@jhu.edu

Smillie, Dan
dsmilli1@jhu.edu

December 19, 2017

Contents

1 Overview

YouTube has a massive collection of videos, and many of them are labeled for search visibility. This dataset is a huge opportunity to do video classification. To facilitate this, Google has (very very recently) released tools to reduce the size of the data and make it accessible to researchers without hundreds of CPU years at their disposal.

We use this data to train a classifier to label videos, demonstrate that some labels are harder to learn than others, and show the models efficacy on wild videos.

2 Dataset

The Youtube-8M dataset is a large-scale labeled video dataset that is compromised of millions YouTube video IDs and labels from over 4700 classes of labels. The data is precomputed so as to reduce the size from Terabytes of data, to merely Gigabytes. This dataset is especially exciting to be working with because of how recently it has been released to the public. Google and YouTube first announced the dataset in September of 2016 and has even released new feature extraction code in November of this year, that we were able to work with.

The dataset comes in two major flavors, video level and frame level features. Despite the fact that over seven million videos have been used to build this set, the fully compressed dataset is still smaller than two terabytes.

The labels, that are coupled with the video IDs, represent different categories of video. For example a video of Lebron James dunking in the middle of a basketball game would receive the labels, sports, game, and possibly celebrity. The average number of labels per video is about three and a half.

The dataset also contained audio features that we did not manipulate in any way but we hypothesized about how coupling those features along with frame and/or video level features could lead to interesting results.

3 Methods

3.1 Models

Google has published support for training Logistic multi-class classifiers for both frame-level and video-level features. Although this is a simple model, we expect it to be sufficient since the data is already preprocessed in a (hopefully) intelligent manner. It would be interesting to see if a shallow neural-network would perform better, but we didn't have the time to reverse-engineer Google's code and create our own model.

Since both frame-level and video-level features have the same (well-studied) model, any differences in the results should reflect the differences in aptitudes of the features, not the models.

3.2 Training

We used ? videos and 3000 steps for both frame-level and video-level features. We saw that the learning rate for both types of features are nearly identical (until...?).

3.3 Evaluation

When we started this project, one of the core things we wanted to show was the difference in ability to learn different labels. This is not built into Googles simple package, so we had to make it ourselves. We used Binary Cross Entropy Loss to evaluate the outputted label probabilities, setting the predicted probability to 0.5% as it seems TensorFlow does not report labels with less than 1% probability. An artifact of this decision is that there is a maximum error for each label, and this maximum is routinely achieved by many of the less common labels.

Using this data, we can generate plots for each label of error vs. iteration, and possibly plot the errors of two labels against each other vs. iteration. The hope of this method is to highlight whether or not some labels train better than other labels. However, Google has some hidden bugs in their code and many of our label error files are basically useless.

We also report the validation error vs. iteration. During our short runs for the presentation, we saw that this error plummets in the first 600 iterations, but we are curious if, by 3000 iterations, it trends upward again. This would be a sign of overfitting and inform us which iteration of the model generalizes best.

4 Results

4.1 Total Error by Iteration

4.2 Compare Label Error

We were able to evaluate the individual label errors at the final iteration. While we are not confident the code ran 100% correctly, there are some patterns in the results which are worth mentioning:

1.

4.3 Label Error by Iteration

Too many bugs to have meaningful results.

5 Conclusions

5.1 Video-Level Features Generalize Best

As we see in the plot of validation error vs. iteration, video-features get more and more superior to frame-features as training time increases. Furthermore, it seems that This means that it more accurately captures the true structure of the labeling problem. In short, the results suggest that video-features are more informative than frame-freatures.

However, neither model reached convergence so it is not proven that video-features are superior. We are only fairly certain in our conclusion, not absolutely certain.

cite err1410.csv - labels 6 and 15 are easier

6 References

YouTube-8M: https://research.google.com/youtube8m/index.html