YouTube Video Classification using Neural Networks on different Distributed Frameworks

Prashant K. Thakur, Paahuni Khandelwal, Heting Wang
Colorado State University

April 11, 2019

1 Introduction

The project uses two most popular framework in machine learning and distributed computing for benchmarking and working with the YouTube-8M video classification dataset.

1.1 TensorFlow

TensorFlow is a framework released by Google for numerical computation using data flow graphs. This framework was developed by the Google Brain Team to facilitate implementing machine learning and deep neural networks research. The data flow graphs comprise of nodes which represent the mathematical operations, and the graph edges representing the flow of data (tensor or multidimensional arrays) in-between the nodes. The framework has C/C++ engine that improves the performance along with a very rich Python API. Distributed TensorFlow provides the flexibility to scale up the system by computing certain portions of the graph on different processes/servers.

1.2 Apache Spark

Apache Spark is an open source framework for cluster-computing. The framework builds a lineage graph and the computation is carried out on different clusters based on the actions. The performance of the framework has been optimized 100 times than Hadoop by using in-memory computation and other optimization. Spark has different packages such as support for SQL queries (*Spark SQL*), data streaming (*Spark Streaming*), graph processing (*GraphX*) and machine learning (*Spark MLLib*). Spark MLLib works on top of Spark Core and has several machine learning algorithms libraries (like - sampling, classification, filtering, clustering, dimensionality reduction, feature extraction etc). The framework has well-supported API for Python, Java, Scala, R.

2 Problem Formulation

2.1 Bench-marking

The scope of this project is to evaluate the benchmark of two most popular distributed frameworks - Apache Spark, Distributed TensorFlow. Apache Spark MLLib has a reach collection of different machine learning algorithms which has been optimized for better performance than MapReduce. Similarly, TensorFlow an open-sourced distributed version to increase the scalability and parallelisms to work on big datasets. In our project, we would deploy these two Frameworks and try to benchmark the performances of Neural Network-based classification of large YouTube Video dataset on clusters. We will measure the performance of the classifier based on the time taken to train the model, a given error rate or loss during training on different nodes (cluster number). We would also try to check the accuracy of the model deployed on these two frameworks and try to analyze the results obtained from the results.

2.2 Dataset

We are working on the labeled dataset from YouTube. We have input attributes (video features and audio features) along with the video id's and our model will try to classify these videos into different class labels, such as politics, wildlife, music, food, hotels and so on. And we have our target attribute 'classes'. Since there could be different topics classes, we are planning to do the classification on selected classes to limit our data size. In order to do so, we have to pre-process the dataset and extract the videos-id and their corresponding features for certain set of classes.

Why Youtube-8M Dataset?

When we search a video on YouTube for given keywords, almost all of the search results have the keyword either in a video title or in a description. However, predicting any video class based on its audio/video features would provide the following benefits -

- Optimize videos filtering regardless of the misleading titles and descriptions.
- Eliminates the term-spam for the videos as audio/video features are used for classifying. Eg. try searching simple query on youtube as " Black Panther full movie". The results are highly viewed videos which are either the trailers or long videos containing links highly possibly a spam.
- Easy implementation of a control mechanism for videos recommendation, parental control for a video channel.
- Use video analysis to categorize any video for detecting emergency conditions using real-time CCTV feeds for traffic control, emergency medical assistance and several other.
- With optimization it can be used to detect Copyright Infringement by correlating different videos.
- Finally, due to the dataset large size, it is well suited to solve this as big data problem.

3 Strategy to solve the problem

3.1 Training and Testing

Our goal is to benchmark both the frameworks. So, we will work upon classification models provided by both the frameworks and works best with them. Therefore, to perform classification on Apache Spark we will be performing Logistic Regression using Neural Network provided in MLLib and in TensorFlow we will develop Convolutional Neural Network (CNN) for classifying videos. Further, we will be performing stratified partitioning which will take into account samples from all the classes while training network. Our model will divide the entire dataset into training, validating and testing sets. We will perform different experiments to get best performance of both the classification models.

3.2 Evaluation Matrix

For evaluating the performance of our classifier, we will use Logarithmic Loss function. Log Loss will measure the performance in terms of accuracy of our classifier on both the frameworks by penalizing false classifications. The log loss increases as the predicted probability diverge from the actual label. So, log loss as 0 means a perfect model. Log Loss assigns a probability to each class between 0 to 1 and can be defined as -

$$-\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log\, p_{ij}$$

where N is the number of instances, M is the total number of possible labels, $\mathrm{y_{ij}}$ is a binary indicator of whether or not label j is the correct classification for instance i, and $\mathrm{p_{ij}}$ is the model probability of assigning label j to instance i.

The Log Loss measures the classifier performance by measuring how much predicted classification varies from the actual label. Although measuring log loss is the better indicator of our model's performance, but we will also measure the accuracy by calculating the correct percentage of classifications made the models.

4 Dataset Description

Youtube 8M dataset - YouTube-8M is a large-scale labeled video dataset that consists of millions of YouTube video IDs and associated labels from a diverse vocabulary of 4700+ visual entities.

Link for dataset-`https://research.google.com/youtube8m/download.html`

This dataset is one of the largest multi-labeled video classification dataset, composed of around 8 million videos(500,000 hours of videos) from past 50 years.The dataset was generated by Video Understanding group within the Machine Perception Research organization at Google using a YouTube video annotation system, which labels videos with their main topics.The labels are machine-generated but they have high-precision and are derived from a variety of human-based signals including metadata and query click signals. The dataset is created in form of tfrecords. So, we can directly read the entire dataset into TensorFlow.

5 Project Timeline

Prashant	Paahuni	Heting	Week
Feasibility Study TensorFlow Framework	Feasibility Study Apache Spark Framework	Analyzing the dataset and extract the required information	Week 1(03/29 - 04/5)
Formulation of the algorithm using TensorFlow inbuilt functions to design software for classification.	Formulation of the algorithm using MLLib to perform classification on Apache Spark.	Performing data filtering and cleaning	Week 2(04/5 - 04/12)
Testing the code on the local machine using the small dataset in TensorFlow	Implementing developed models to train on the small dataset on Apache Spark	Evaluate the models on both the frameworks in cluster	Week 3(04/12 - 04/19)
Verify the results and improve the method to reduce errors and start with writing the report	Improve the efficiency on Spark Framework and prepare presentation	Write the project report	Week 4(04/19 - 04/26)

6 Bibliography

1. https://research.google.com/youtube8m/
2. https://www.tensorflow.org/tutorials/deep_cnn
3. http://spark.apache.org/docs/1.2.1/mllib-classification-regression.html
4. https://arxiv.org/pdf/1609.08675.pdf
5. http://www.exegetic.biz/blog/2015/12/making-sense-logarithmic-loss/