Predicting Video Tags Using Google's YouTube-8M Dataset

Tatyana Li
litatyan@msu.edu

1. INTRODUCTION

On September 2016, Google announced a release of the large labeled dataset, YouTube-8M, for video understanding research. The dataset contained about 8 million YouTube videos, amounting to more than 500K hours of video content. Earlier this month, **Google hosted a Google Cloud and YouTube-8M Video Understanding Challenge** on a Kaggle platform, providing an updated version of the dataset with new labels and newly added audio features. This project applied several classification algorithms to the newly released You-Tube V2. dataset with the purpose to accurately assign labels to a video. Specifically, for video-level features, for each entity e, we trained one-vs-all independent binary logistic classifiers and a mixture of experts (ME) model. The results indicated that a ME model with, both, video- and audio features, outperforms logistic regression classfiers. Prior work to date suggested that deep neural networks outperform other models for image classification. Similarly, in this project, the LSTM model trained on frame-level video features provided the best generalization performance on a test set, achieving global average precision score of 0.791.

2. PROBLEM STATEMENT

The purpose of this project is to implement and compare classification algorithms to predict video labels based on the video and frame-level features provided. Ultimately, as part of the Kaggle competiton, we would like to identify the classification model that gives the best label predictions based on the test data provided by Google research.

3. RELATED WORK

Until very recently, the research on image recognition had been applied to small labeled image datasets, e.g. Caltech 101/256 [1], PASCAL [2]. Due to a rapid increase in computational power, there has been a significant advancement in image understanding research employing much larger-scale datasets, e.g. ImageNet [3] and SUN [4]. As an example, Krizhevsky et al. [5] performed training of a large, deep convolutional neural network to classify a subset of 1.2M images from the ImageNet database into 1000 different classes. The deep neural network contained 60M parameters and 650,000 neurons, with 5 convolutional layers. The researchers showed that using purely supervised learning, the deep convolutional neural network is capable of producing high classification accuracy. The authors report top-1 and top-5 error rates of 37.5% and 17%, which was significantly lower, compared to the previously established benchmark. Sermanet et al. [6] used convolutional neural nets for digit classification of house numbers from SVHN classification dataset, containing 32x32 images, with three color channels. The authors report that with L4 pooling, they achieved a state-of-the-art performance, with classification accuracy close to 95%.

Although, relatively less work has been done on large-scale video classification, compared to the image domain, still, significant progress has been made in the area of video understanding research. The scale of the datasets progressed from smaller labeled video datasets Hollywood 2 [7], Weizmann [8], containing only several thousands of video clips to much larger-scale datasets, such as YFCC-100M dataset [9] with 800K videos and metadata, containing video titles, descriptions and tags; ActivityNet [10], which is a large-scale video benchmark with several thousand videos and 200 human activity classes, and can be used to compare algorithms for human activity understanding. As of today, the Sports-1M [11] has been considered one of the largest video datasets, containing around 1M video instances. To date, however, there has been no video datasets, comparable in scale and diversity to YouTube-8M. The updated YouTube-8M dataset contains over 8M videos and 1.9B video frames and provides the richest resource for research in video understanding and representation learning. The dataset comes with ”pre-computed state-of-the-art features for 1.9 billion video frames”, which ”levels the playing field” for researhers to utilize the dataset on a scale never seen before.

4. DATA

YouTube-8M dataset represents a benchmark in video understanding, with the key purpose to identify the main topics associated with a video. The data available for the Kaggle competition is a second version of the dataset, YouTube-8M V2. The dataset contains YouTube videos from a diverse range of categories from gaming and entertainment to sports, foods, health, science and education. It sets this dataset apart from its' predecessors, where, for the latter, categories were limited to a single domain, such as, for example, sports or action. The Google researchers used a video annotation system to produce topic annotations, and then obtain videos for a certain topic. Overall, the YouTube-8M V2. dataset contains 4716 classes with over 7M videos. A given video may have more than one class label, with an average of 3.4 labels per video. The train set, released by Google, contains over 5M video examples, validation set contains over 1M videos, and test set contains 700,640 videos.

The dataset comes with standard frame- and video-level features. Still, there seems to be a lot of room for experimenting with video-level representation learning approaches.

To visualize the distribution of the labels, a subsample was drawn from the avaialble training set of over 7M videos. For each sample in the training set, the number of labels assigned to it were counted. For this subset of training data, we can estimate the true distribution of number of labels. The graph is presented below in Figure 1. We observe that the majority of samples have relatively low number of labels. Based on the subsample of the training set, Figure 2 below shows the top 20 labels by occurence in the subsample drawn from the train data.

Figure 1: Frequency distribution of labels.

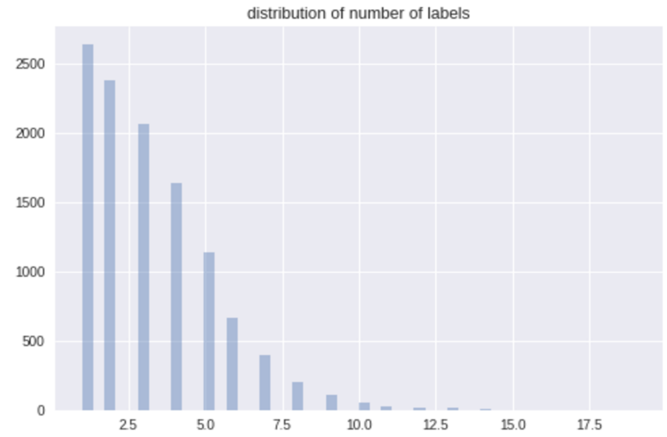
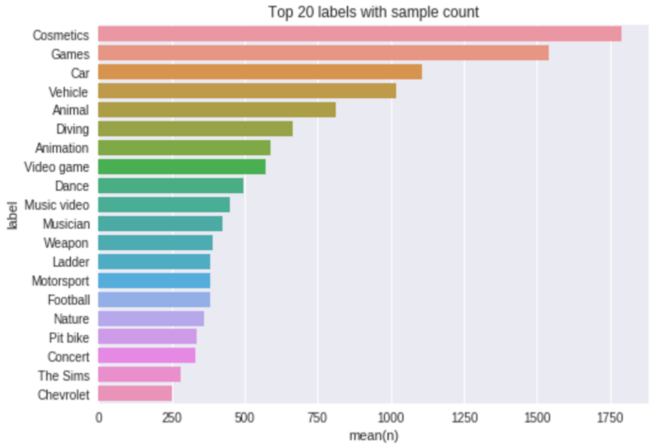


Figure 2: Frequency of top 20 labels.



As part of the Kaggle competition, the dataset comes with already pre-extracted and pre-processed features. In particular, each video was decoded at 1-frame-per-second up to the first 6 minutes. Then, the decoded frames went through the Inception network trained on ImageNet. This results in static frame-level features. Further, principal component anaysis (PCA) and whitening were applied for dimensionality reduction; and quantization had been applied. For both, video and frame level features, there are 1024 and 128, video and audio features, respectively.

5. MODELS

Below, we describe the technical details of the models used in this project.

Logistic Regression We define the logistic regression model as follows:

$$ln\frac{p(\mathbf{x}_i)}{1-p(x_i)} = \mathbf{w}^T\mathbf{x}_i \ , \ i = 1, 2, ..., N$$

where:

$\mathbf{x}_i$ is a d-dimensional feature vector, $\mathbf{x}_i \in \mathbb{R}^d$
$\mathbf{w}^T$ is a d-dimensional weight vector, $\mathbf{w} \in \mathbb{R}^d$
$p(\mathbf{x}_i) = \sigma(\mathbf{w}^T\mathbf{x}_i) = \frac{1}{1+exp(\mathbf{w}^T\mathbf{x}_i)}$

Thus, the log-likelihood function is defined as:

$$l(\mathbf{w}) = \sum_{i=1}^{N}\Big\{y_i\, lnp(\mathbf{x}_i) + (1-y_i)ln(1-p(\mathbf{x}_i))\Big\}$$

We, then, learn the weight vector $\mathbf{w}$, by minimizing the following cross-entropy function $E(\mathbf{w})$ with $l2$-regularization penalty to discourage large weights:

$$-\sum_{i=1}^{N}\Big\{y_i\, ln(p(\mathbf{x}_i)+\Delta) + (1-y_i)ln(1-p(\mathbf{x}_i)+\Delta)\Big\} + ||\mathbf{w}||_2^2$$

The Δ is a hyperparameter and is a small positive constant which we add to avoid numerical difficulties when taking a logarithm of zero values.

Mixture of Experts (ME)
In the past 20 years, ME models have been found useful in combination with many current classification and regression algorithms because of their flexibility and modular structure. The ME model is competitive for nonlinear classification problems with data that contain natural distinct subsets of patterns, among others. The original ME model was developed by Jacobs et al. [12], and can be viewed as a tree-sructured architecture, having three main components: several experts that are either regression functions or classifiers; a gate that makes soft partitions of the input space and defines those regions where the individual expert opinions are trustworthy; and a probabilistic model to combine the experts and the gate. The model is a weighted sum of experts, where the weights are the input-dependent gates. More concretely, in the ME architecture, a set of experts and a gate "cooperate" with each other to solve a nonlinear supervised learning problem by dividing the input space into a nested set of regions (see Figure 1). Figure 3 below illustrates a simplified classification example for ME.

The red and blue points indicate two different classes and present a nonlinear classification example. The gate softly partitions the regions, where the individual experts' opinion are trustworthy, such that, to the right of the gating line, the first expert is responsible, and to the left of the gating line, the second expert is responsible. With this divide-and-conquer approach, the snonlinear classification problem has been simplified to two linear classification problems. Figure 4 below presents an example of a typical ME architecture for classification problem.

Figure 3: Simplified classification example for ME.

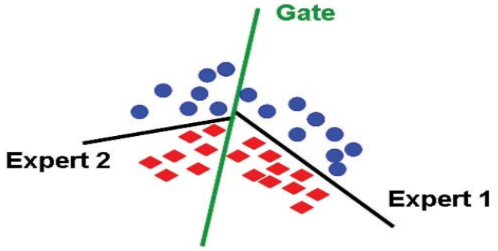
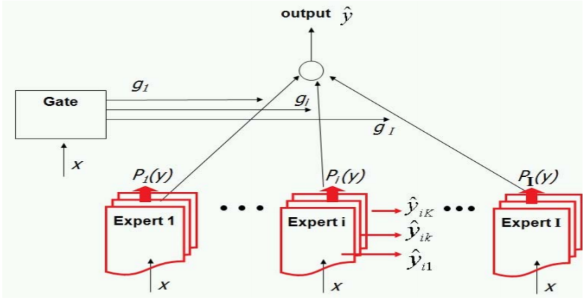


Figure 4: typical ME architecture for classification.



Given, the i-th expert, for each entity e (class label), we use a sigmoid function to model the probability of an entity:

$$\sigma_i(\mathbf{w_i^T x}) = \frac{\mathbf{1}}{\mathbf{1+exp(-w_i^T x)}}$$

The gating network is also a generalized function, and its' i-th output $g(x, v_i)$ is a softmax function:

$$g(x, v_i) = \frac{exp(v_i^T \mathbf{x})}{\sum_{k=1}^{N} exp(v_k^T \mathbf{x})}$$

where: v_i is a weight vector.

Then, the overall output of the ME architecture is:

$$\hat{p}(x) = \sum_{k=1}^{N} g(\mathbf{x}, \mathbf{v_k})\sigma_{\mathbf{k}}(\mathbf{w_i^T x})$$

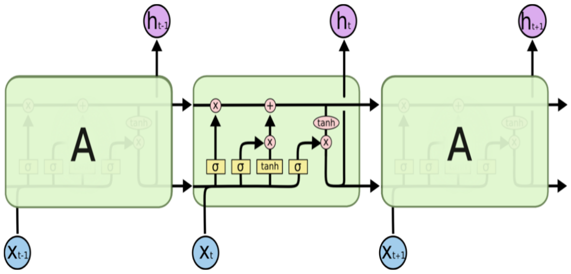
Given, a set of training examples, $\{\mathbf{x_i}, \mathbf{y_i}\}_{\mathbf{i=1,...,N}}$ for a binary classifier, where $\mathbf{x_i}$ is a feature vector and y_i is a ground truth, then $L(p_i, y_i)$ be the log-loss between the predicted probability and the ground-truth:

$$L(p, y) = -ylogp - (1-y)log(1-p)$$

Thus, we minimize this above mentioned loss function with respect to the softmax weight $\mathbf{v}$ and a logistic weight $\mathbf{w}$.

Recurrent Neural Network (LSTM)
Long Short Term Memory (LSTM) network is a special kind of network capable of modeling long-term dependencies in the data, by "remembering" information for long periods of time. Unlike in the standard recurrent neural network (RNN), the repeating module in LSTM has a different structure. Instead of having a single neural network layer, there are four layers, that interact in a particular way. Figure 5 below demonstrates a typical LSTM architecture.

Figure 5: Typical LSTM architecture.



The LSTM calculates a hidden state s_t as follows:

$$\begin{aligned} i &= \sigma(x_t U^i + s_{t-1} W^i) \\ f &= \sigma(x_t U^f + s_{t-1} W^f) \\ o &= \sigma(x_t U^o + s_{t-1} W^o) \\ g &= tanh(x_t U^g + s_{t-1} W^g) \\ c_t &= c_{t-1} * f + g * i \\ s_t &= tanh(c_t) * o \end{aligned}$$

where:
i, f, o are input, forget and output states
g is a "candidate" hidden state, computed based on the current input and previous hidden state
c_t is an internal memory of the unit

Given, the memory c_t, the hidden state s_t is computed, by multiplying the memory with the output gate.

6. EXPERIMENTS

5.1 EVALUATION METRICS

The evaluation metrics used to assess model performance were generated by Google/Kaggle and are defined as follows:

Hit@k A fraction of test samples that contain at least one of the ground truth labels in the top k predictions. If $rank_{v,e}$ is the rank of entity e on video v (with the best scoring entity having rank 1), and G_v is the set of ground-truth entities for v, then Hit@k can be written as:

$$\frac{1}{|V|} \sum_{v \in V} V_{e \in G_v} \mathbb{I}(rank_{v,e} \leq \ k), \qquad (1)$$

where: V is logical OR

Precision at equal recall rate (PERR) The video-level annotation precision was measured when the same number of entities per video are retrieved as there are in the ground-truth. With the same notation as for Hit@k, PERR can be written as:

$$\frac{1}{|V:|G_v|>0|}\sum_{v\in V:|G_v|>0}\left[\frac{1}{|G_v s|}\sum_{e\in G_v}\mathbb{I}(rank_{v,e}\leq\ |G_v|\right] \quad (2)$$

Average Global Precision (AGP), average precision across all predictions and all videos:

$$GAP=\sum_{i=1}^{N}p(i)\Delta r(i) \quad (3)$$

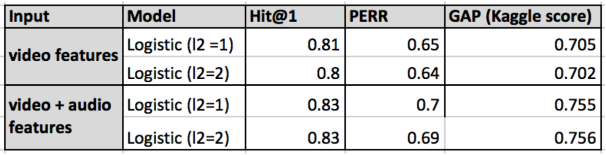
where:

N = number of final predictions for all videos
$p(i)$ is precision and
$\Delta r(i)$ is recall

5.2. VIDEO-LEVEL FEATURES

Logistic Regression. For each entity e (label), we trained 4716 independent binary logistic regression classifiers. We used Adam optimizer, as it works well in practice and compares favorably to other adaptive learning algorithms, e.g. AdaGrad and set the learning rate to 0.01. We experimented with two types of features and two values of regularization parameter. Each logistic model was trained on the Google Cloud for a total of about 30 minutes, validation of the model was run for 20 minutes. Afterwards, the model was tested on 700,640 test examples and predictions were generated. The Table 1 below presents the results of the experiments with logistic models using only video features as predictors, as well as, both, video and audio features. For each set of features, we experimented with two values of the regularization parameter $\lambda = 1$ and $\lambda = 2$.

Table 1: Logistic regression results (video-level)

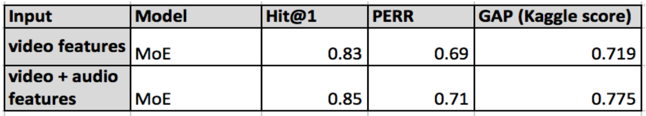


Input	Model	Hit@1	PERR	GAP (Kaggle score)
video features	Logistic (l2 =1)	0.81	0.65	0.705
	Logistic (l2=2)	0.8	0.64	0.702
video + audio features	Logistic (l2=1)	0.83	0.7	0.755
	Logistic (l2=2)	0.83	0.69	0.756

We observe that the results for the logistic models with video and audio features with $\lambda = 1$ and $\lambda = 2$ penalties are comparable to each other. Specifically, the values of Hit@1 at the end of training were 0.83 and PERR was around 0.70 for both models. The GAP (or Kaggle score) for the model with regularization parameter $\lambda = 2$ was 0.756, which is slightly higher than for the model with $\lambda = 1$, which was 0.755, respectively. However, clearly, the logistic models with, both, video and audio features perform noticeably better, compared to the models without audio features. All evaluation metrics for the latter were somewhat lower: Hit@1 around 0.80 and PERR was around 0.65. The GAP Kaggle score for logistic model with regularization parameter $\lambda = 1$ is 0.705, while for the model with $\lambda = 2$ is 0.702.

Mixture of Experts. We next experimented with mixture of experts (ME) models. Similar to the logistic regression, we trained separate 4716 mixture of logistic regression experts models, first, using only video features and then, both, video and audio features. Each ME model was trained for a period of around 30 minutes, validation was performed for 20 minutes. After that, the trained model was used on a test set of 700,640 test instances to generate predictions for Kaggle submission. For ME models, we used two mixtures, l2-norm regularization penalty equal to 1e-8 and Adam optimization algorithm. Table 2 below presents the results of the experiments.

Table 2: Mixture of Experts results (video-level)



Input	Model	Hit@1	PERR	GAP (Kaggle score)
video features	MoE	0.83	0.69	0.719
video + audio features	MoE	0.85	0.71	0.775

Not surprisingly, the ME model with two sets of features, video and audio, outperforms the model with video features only. While, the Hit@1 and PERR training metrics are relatively comparable between the two, around 0.8 and 0.7, respectively, the gain in global average precision by using ME with full set of features is non-trivial, GAP = 0.775. Moreover, the mixture of experts model delivers better generalization performance on a test set, compared to independent logistic classifiers.

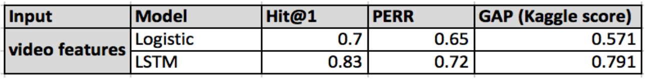
5.3. FRAME-LEVEL FEATURES

Logistic Regression
Due to extensive computational time required to train models on frame level features, limited experiments have been performed. For each entity (class label) e, we trained 4716 one-vs-all binary logistic classifiers over the average of frame level features. Even on the Google Cloud, the training is computationally expensive. The training of the model was performed for approximately 4 hours, validation and inference on the test set took in total of around 2 hours. We used Adam optimizer, with l2 regularization penalty of 1, and a learning rate of 0.0002. The results are presented in Table 3 below. It was expected, that the results of the logistic model with frame-level features would be comparable to the logistic model with video-level features. However, the model with frame-level inputs peformed poorly and gave the worst generalization performance on the test set in terms of GAP = 0.571. It is possible, though, that this model requires longer period of training to achieve accuracy, comparable to video-level models.

Recurrent Neural Network (LSTM).
We also trained LSTM with two stacked layers, 1024 hidden units, and a softmax as an ouput layer. Since, we didn't have access to raw video frames, we could only train the LSTM and the softmax layers. Linearly increasing per-frame weights were used, starting from 1/N to 1 for the last frame. The results on the evaluation metrics are also given in Table 3. We observe that, using video-level features, the LSTM model outperforms all other models considered, giving GAP close to 0.80. Moreover, it also performs better than logistic or ME models, which use, both, video and audio features.

Due to extensive computational time required for training, we didn't train the models with full set of frame-level features, including, both, video and audio.

Table 3: Logistic regression and LSTM results (frame-level)



Input	Model	Hit@1	PERR	GAP (Kaggle score)
video features	Logistic	0.7	0.65	0.571
	LSTM	0.83	0.72	0.791

7. CONCLUSIONS AND FUTURE WORK

Due to the time constraints of the project and extensive computational time required to train models with frame-level features, the scope of the experiments performed within this project is limited. The main purpose of this project was to use a noisy real-world dataset available as a Kaggle competition to address a practical problem. However, due to the recent release of the dataset by Google and the novelty of the data, being one of the largest multi-label classification datasets available to date, this project contributes to setting the benchmarks in video representation learning. As expected, the neural network model resulted in the best generalization performance on frame-level features, and it is likely that a certain type of deep learning model will result in an overall winning score in the Kaggle competition. Also, unsurprisingly, the mixture of experts models using video-level features performed comparatively well.

One obvious side effect of having millions of training examples is expensive computation. Due to this, we couldn't perform hyperparameter tuning for any of the models. Most of regularization parameters, learning rates or other hyperparameters were set by simple experimentation or rules of thumb. Future work can address this. In a similar vein, more experimentation with neural network architectures could be performed which would likely result in a better performing model. We used Adam optimization algorithm for weights learning, as it generally works well in practice. However, we haven't experimented with other optimizers which could potentially provide faster learning etc. As the time required for training is one of the major bottlenecks in this competition, one important lesson learned is to find a model and optimizer which would learn efficiently (i.e. fast).

Lastly, this is a multi-label classification problem. Future work may also focus on models that take into the account the structure of the label information.

8. REFERENCES

[1] G. Griffin, A.Holub, and P.Perona. Caltech-256 object category dataset. technical report 7694, california institute of technology,. 2007.

[2] M. Everingham, V.V. Cool, C.K.I. Williams, J.Winn, and A.Zisserman. The pascal visual object classes (voc) challenge,. 2009.

[3] J.Deng, W.Dong, R.Socher, L.Jia Li, K.li, and L.Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*, 2009.

[4] J.Xiao, K.A.Ehinger, J.Hays, A.Torralba, A.Olivia, and J.Xiao. Sun database: Exploring a large collection of scence categories. 2013.

[5] A. Krizhevsky, I.Sutskever, and G.E.Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems, NIPS*, pages 1097–1105, 2012.

[6] P. Sermanet, S.Chintala, and Y.LeCun. Convolutional neural networks applied to house numbers digit classification. In *ICPR*, 2012.

[7] I.Laptev, M.Marszalek, C.Schmidt, and B.Rozenfeld. Learning realistic human actions from movies. In *Conference on Computer Vision and Pattern Recognition, IEEE*, 2008.

[8] M.Blank, L.Gorelick, E.Shechtman, M.Irani, and R.Basri. Actions as space-time shapes. In *Proceedings of the International Conference on Computer Vision, ICCV*, 2005.

[9] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning, ICML*, pages 448–456, 2015.

[10] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–970, 2015.

[11] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1725–1732, Columbus, Ohio, USA, 2014.

[12] M.I. Jordan. Hierarchical mixtures of experts and the em algorithm. In *Neural Computation, 6*, 1994.