Boosting Up Segment-level Video Classification Performance with Label Correlation and Reweighting

Lei Wang *, Song Chen *, Hui Zhou *
Autohome Corporation, Beijing, China
wlspe35@gmail.com

Abstract

This paper introduces a solution to the 3rd Youtube-8M video understanding challenge. The main focus of the solution is to analyze the label dependencies of multi-label videos and explore this important information when the ground-truth label is incomplete. Our final solution consists of a base model which is a mixture of NeXtVLAD and GRU models, a reweight label matrix and a label correlation matrix. The final solution got a MAP 0.782 score on the private leaderboard. We also do some research on the GCN (Graph Convolutional Network) and add the GCN network into the video classification task.

1. Introduction

In Recent years, video has become a more and more popular form of entertainment. Huge amount of video data had been produced and saved. Youtube-8m, the largest multi-label video classification dataset provided by Google, composed of 6.1Million videos (350k hours of video), annotated with a vocabulary of 3862 visual entities. The Dataset provides CNN-Pretrained frame-level visual and audio features of each video for multi-label video classification. This year, an extra segment-level dataset composed of 237K human-verified segment labels for 1000 classes had been provided which encourage participants to localize video-level labels to the precise time in the video where the label actually appears. Thus, the main focus of this year's completion is how to leverage noisy video-level labels and a small subset of segment-level set jointly in order to better annotate and temporally localize concepts of interest in videos.

In this paper, we build label reweighting matrix and label correlation network to improve video classification and get good performance based on the results.

* Authors contributed equally to this work

2. Related Works

NeXtVLAD [1], proposed in last year's competition, was proved to be an efficient and fast method to classify videos. Inspired by the method of ResNeXt, the author successfully decomposed the video feature vector with high-dimension into a group of low dimension vectors. This network significantly reduced the parameters of previous NetVLAD network, but still got remarkable performance on feature aggregating and large-scale video classification.

RNN [2] has been proved to perform excellently when modeling sequential data. Researchers commonly use RNN to model temporal information in videos which CNN network hard to capture. GRU [3] is an important ingredient of RNN architectures, which can avoid the problem of gradient vanishing. Attention-GRU [4] refers to GRU with the attention mechanism, which helps to distinguish the influence of different features on the current prediction.

In order to combine the spatial features and temporal features of video tasks, Two Stream CNN Networks [6], 3D-Convolution Networks [7] are proposed. These models also showed good performance on video understanding task
.

3. Our Approach

We firstly show the model architecture used in this task, then we present our research on label dependencies of multi-label videos.

3.1 Model Architecture

The final model is a mixture of 3 NeXtVLAD models and one GRU model. 3 NeXtVLAD models are configured with different parameters which intend to improve the feature aggregating ability. A GRU network branch had been added into the network to help model learn more temporal information.

As shown in Figure 1, the prediction of model is a mixture of the 4 models in the graph. The weighting of this 4

models would be trained form the network. Table 1 shows the detail parameters of the 3 NeXtVLAD models.

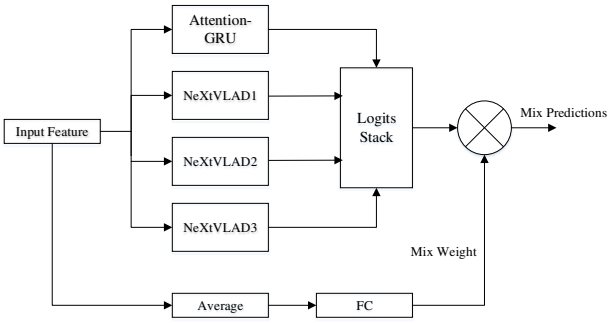

Figure 1. The architecture of our model

	Group	Cluster Size	Hidden Size	Reduc-tion
NeXtVLAD1	8	112	2048	16
NeXtVLAD2	8	136	2048	16
NeXtVLAD3	8	112	2048	8

Table 1. Parameters of 3 NeXtVLAD Models

As shown in Figure 2, an attention network had been added to the GRU network to improve the temporal localization ability of the model.

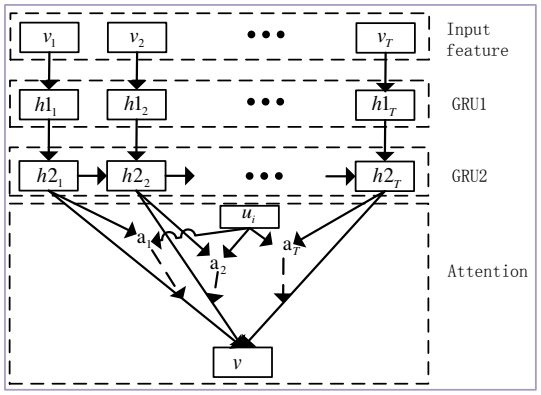

Figure 2. The architecture of Attention

3.2 Label Reweighting

The Youtube-8M video classification task is a multi-label classification task [8] [9]. But for this year's segment-level dataset, the annotated data only labeled 0 or 1 for only one class. This one-hot like label score will treat other classes that not annotated all be 0. But the ground-truth score of a video segment for classes not annotated may be different.

We proposed a method which give a large weight for the annotated class and give a small weight for the classes that not annotated when calculate loss [13] . This weighted cross-entropy method will help the model to learn better from the incomplete dataset.

The original binary cross entropy loss function is:

$$loss = -y * \log(p) - (1-y) * \log(1-p) \tag{1}$$

where $y \in R^{B \times C}$ is segment label matrix, $p \in R^{B \times C}$ is model predictions, B is batch-size, C is the number of all classes. The new loss function is as follows:

$$loss\_label\_reweighted = w * \text{loss} \tag{2}$$

where $w \in R^{B \times C}$ is defined by

$$w_{ij} = \begin{cases} m & \text{if category j be labeled} \\ n & else \end{cases} \tag{3}$$

n is the small weight added to classes not annotated, *m* is the large value weight for the annotated class.

3.3 Label Correlation Matrix

In our multi-label classification task, for a video, certain labels may occur together with high probability and some other labels may never appear at the same time. So analysis of the label relationship help us to improve performance especially when the segment label is incomplete [16].

In this year's competition, we use conditional probability between labels to incorporate label correlation information into our modeling [11]. Based on the 2nd Youtube-8M dataset, first we count the occurrence times of each label, a matrix $N \in R^C$ will be obtained. C is the number of all categories. Then we count the concurring times of all label pairs, so we can get the concurring matrix $M \in R^{C \times C}$. The final conditional probability matrix $P \in R^{C \times C}$ is calculated by:

$$P_{ij} = \frac{M_{ij}}{N_i} \tag{4}$$

P_{ij} means the occurrence probability of label *j* when label *i* appears (label *i* scores 1.0 in the annotation data).

However, the co-occurrence matrix from training and test dataset may not be completely consistent and the dataset labels are not 100% correct. On the other hand, some rare co-occurrence pairs may not be true relationship for us but just noises. Thus, we can use a threshold τ to filter P to

avoid the negative effect by the weakly correlated labels. The function can be written as:

$$P'_{ij} = \begin{cases} P_{ij} & \text{if } P_{ij} \geq \tau \\ 0 & else \end{cases} \quad (5)$$

So for a segment label score matrix *y*, we will change the matrix value by:

$$y' = y + P' \quad (6)$$

By using the label correlation matrix, when a segment of video annotated 1.0 for a certain label, we will give a score (filtered by the threshold value) to its related labels instead of setting all other classes 0. We think it is useful to do this, especially when the ground-truth label is incomplete.

3.4 Graph Conluvtional Network (GCN)

GCN [18] was introduced to perform semi-supervised classification. The basic idea is to update the node representations by propagating information between nodes.

For multi-label video classification task, the label dependencies is an important information. In our task, each label will be a node of the Graph, the line between two nodes indicates their relationship [15] [16]. So we can train a matrix which indicates the relationship of all nodes. Different from the label correlation matrix which discussed in section 3.3, the matrix here is trained by model instead of a statistics of the dataset.

Take a simplified label correlation graph extracted from our dataset as an example, label BMW--->Label Car means when BMW label appears, label Car is likely to happen, but the reverse may not be true. Label Car has high relation with all other labels, label pairs with no line connected indicates this two labels have no relationship from each other.

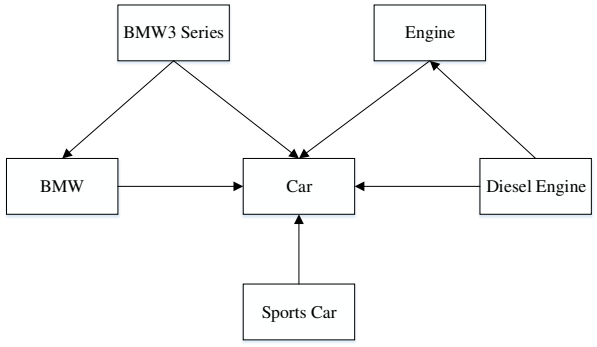

Figure 3: A sample graph of the objected labels

The GCN network implementation is shown as Figure 4. The GCN module consists of two layers of stacked GCNs, GCN1 and GCN2, which help to learn the label graph to map these label representations into a set of inter-dependent classifiers.

$\tilde{A}$ is the input correlation matrix which initialized by the value of matrix P which mentioned in Section 3.3. W_1 and W_2 are the matrix which would be trained in the network. W is the generated classifiers help to do the classification.

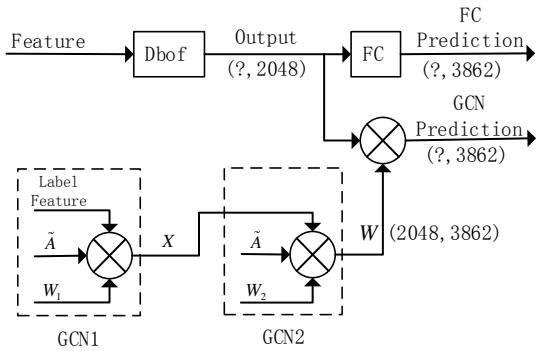

Figure 4. DBOF with GCN network

3.5 Gaussian noises

This year's dataset is very small. To avoid over-fitting when training models, we add randomly generated Gaussian noises and randomly injected into each element of the input feature vector.

As Figure 5 shows, noise will be added to the input feature vector, the mask vector randomly select 50% of dimension and set the value to 1. The Gaussian noise here is independent but with the same distribution for different input vectors.

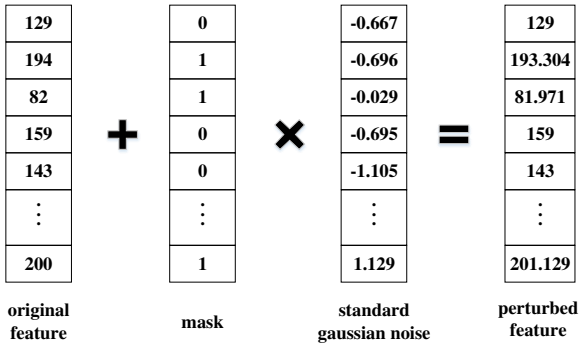

Figure 5. Gaussian noise with mask

4. Experiments

4.1 Evaluation

Model performance is evaluated according to the Mean Average Precision @ K (MAP@K), where K=100,000.

$$MAP@100,000 = \frac{1}{C}\sum_{c=1}^{C}\frac{\sum_{k=1}^{n} P(k)\times rel(k)}{N_c} \qquad (7)$$

where C is the number of classes (1000), n is the number of segments predicted for each class. N_c is the number of positively-labeled segments for each class. $P(k)$ is the precision for top K predicted segments. $rel(k)$ is a function to judge if the segment at rank k is corrected predicted (equaling 1 if correct else 0).

4.2. Training details

Here we introduce how we did our training experiments. First, pre-train our model on the video-level training dataset with an initial learning rate of 0.0002 and a batch size of 80 for 200k steps. Second, fine-tune model on the segment-level dataset with an initial learning rate of 0.00001 and a batch size of 80 for 250k steps.

For the fine-tune step, we split the segment-level dataset into training (80%), validation (20%) for each class. We use the validation part to evaluate whether a method is worked and whether should be added to the final model. For the final submission, we fine-tuned the optimized model on the whole segment-level dataset.

Different model architectures (include GCN network) were experimented both on the pre-train stage and the fine-tune stage. Label reweight matrix and label correlation matrix were added to the model only on the fine-tune stage. Gaussian noises were also used only on the fine-tune stage.

The map score listed below were based on the validation part of the segment-level dataset. A combination of NeXtVLAD and GRU mixed model, label weighting and label correlation matrix got a 0.790 score on the public leaderboard and 0.782 on the private leaderboard.

4.3 Results

4.3.1 Model architecture results

As shown in Table 2, we compared the performance of different model architectures. The model with 3 different NeXtVLAD models and 1 Attention-GRU model showed better performance than the other two architectures. The final model got a map@{100000} score 0.778.

Model	map@{100k}
3NeXtVLAD_Same	0.770
3NeXtVLAD_Diff	0.773
3NeXtVLAD_Diff + Attention-GRU	0.778

Table 2. Different model architecture results.

4.3.2 Label reweight results

Table 3 shows our experiment results on label reweighting. We only tested label reweight in the fine-tune stage with different parameters but a same pre-trained model.

The Model_0.778 refers to the model mentioned above which scored 0.778. The label reweight method improved score from 0.778 to 0.782 and we selected n=0.1, m=2.5 as our final parameters based on the result.

Model	map@{100k}
Model_0.778 + reweight(n=0.1,m=2.0)	0.782
Model_0.778 + reweight(n=0.1,m=2.5)	0.783
Model_0.778+ reweight(n=0.1,m=3.0)	0.781

Table 3. Label Reweight experiment results.

4.3.3 Label correlation matrix results

Table 4 shows our experiment results on label correlation matrix. By using the label correlation matrix, ground-truth label score changed for related labels. We use the new label score to calculate loss in our loss function. We only test the label correlation matrix in the fine-tune stage with different parameters. All experiments here are based on a same pre-trained model.

Model_0.783 refers to the model scored 0.783 which mentioned in last section. The result showed that label correlation matrix improves the score most in all our experiments.

Model	map@{100k}
Model_0.783 + Label Correlation(τ set 0.4)	0.788
Model_0.783 + Label Correlation(τ set 0.5)	0.791
Model_0.783 + Label Correlation(τ set 0.6)	0.789

Table 4. Label Correlation experiment results

4.3.4 GCN results.

Due to time and resource constraints, we only tried GCN on baseline DBOF model.

As shown in Table 5, the result showed that the GCN generated classifiers helped to improve the performance compared with a single DBOF model.

Model	map@{100k}
Dbof	0.740
Dbof with GCN	0.749

Table 5. GCN experiment results.

5. Conclusion and Future Work

In this paper, we presented our solution to the 3rd YouTube-8M Challenge. We found label dependency information is useful for multi-label video classification task especially when ground-truth dataset is incomplete. Experiments showed label reweight and label correlation matrix would improve the performance of video classification. We would like to explore more label dependencies in the future. GCN network also proved to be useful in this task, we think it deserves us to do more experiments on combining GCN network with other state of the art video classification networks.

References

[1]. Rongcheng Lin, Jing Xiao, Jianping Fan: NeXtVLAD: An Efficient Neural Network to Aggregate Frame-level Features for Large-scale Video Classification. In: ECCV, workshop(2018)

[2]. Jeffrey L Elman. Finding structure in time. Cognitive science, 14(2):179–211, 1990

[3]. Kyunghyun Cho, Bart Van Merrienboer, ¨ Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv, 2014.

[4]. Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. Attention-based models for speech recognition. In NIPS, pages 577–585, 2015.

[5]. Junyoung Chung, CaglarGulcehre, KyungHyun Cho, YoshuaBengio: Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. In: NIPS, Deep Learning and Representation Learning Workshop (2014)

[6]. Karen Simonyan, Andrew Zisserman, Two-Stream Convolutional Networks for Action Recognition in Videos. In: NIPS (2014)

[7]. Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, Manohar Paluri Learning Spatiotemporal Features With 3D Convolutional Networks. In: ICCV(2015)

[8]. Huang Y, Wang W, Wang L, et al. Multi-task deep neural network for multi-label learning[C]//2013 IEEE International Conference on Image Processing. IEEE, 2013: 2897-2900.

[9]. Tsoumakas G, Katakis I. Multi-label classification: An Overview [J]. International Journal of Data Warehousing and Mining (IJDWM), 2007, 3(3): 1-13.

[10]. Wang H D, Zhang T, Wu J. The monkey typing solution to the youtube-8m video understanding challenge [J]. arXiv preprint arXiv:1706.05150, 2017.

[11]. Chen Z M, Wei X S, Wang P, et al. Multi-Label Image Recognition with Graph Convolutional Networks[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019: 5177-5186.

[12]. Zhou Z H, Zhang M L. Multi-instance multi-label learning with application to scene classification[C]//Proceedings of the 19th International Conference on Neural Information Processing Systems. MIT Press, 2006: 1609-1616.

[13]. Panchapagesan S, Sun M, Khare A, et al. Multi-Task Learning and Weighted Cross-Entropy for DNN-Based Keyword Spotting[C]//Interspeech. 2016: 760-764.

[14]. Lin T Y, Goyal P, Girshick R, et al. Focal loss for dense object detection[C]//Proceedings of the IEEE international conference on computer vision. 2017: 2980-2988.

[15]. Zhang Z, Sabuncu M. Generalized cross entropy loss for training deep neural networks with noisy labels[C]//Advances in neural information processing systems. 2018: 8778-8788.

[16]. Pereira R B, Plastino A, Zadrozny B, et al. Correlation analysis of performance measures for multi-label classification [J]. Information Processing & Management, 2018, 54(3): 359-369.

[17]. Thomas N. Kipf, Max Welling: Semi-Supervised Classification with Graph Convolutional Networks. In: ICLR (2017)

[18]. Zhao-Min Chen,Xiu-Shen Wei, Peng Wang,Yanwen Guo: Multi-Label Image Recognition with Graph Convolutional Networks. In: CVPR (2019)

[19]. Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, Eduard Hovy: Hierarchical Attention Networks for Document Classification. In:NAACL-HLT (2016), pages 1480–1489

[20]. Thomas N. Kipf, Max Welling: Semi-Supervised Classification with Graph Convolutional Networks. In: ICLR (2017)