Approach for Video Classification with Multi-label on YouTube-8M Dataset

Kwangsoo Shin*[0000−0002−8511−8838], Junhyeong Jeon*[0000−0002−8151−8375], Seungbin Lee[0000−0002−1917−0224], Boyoung Lim, Minsoo Jeong, Jongho Nang

Department of Computer Science and Engineering, Sogang University
{ksshin,junhyeong.jeon,mercileesb,bylim,msjeong,jhnang}@sogang.ac.kr
http://mmlab.sogang.ac.kr

Abstract. Video traffic is increasing at a considerable rate due to the spread of personal media and advancements in media technology. Accordingly, there is a growing need for techniques to automatically classify moving images. This paper use NetVLAD and NetFV models and the Huber loss function for video classification problem and YouTube-8M dataset to verify the experiment. We tried various attempts according to the dataset and optimize hyperparameters, ultimately obtain a GAP score of 0.8668.

Keywords: Video classification · Large-scale video · Multi-label

1 Introduction

Video traffic from video sites such as YouTube has increased in recent years. The growth of personal media through technological development is particularly remarkable. With the development of smartphones, media is now brought to the consumer's hand, and individuals are no longer only consumers of multimedia but are now producers as well. This trend may be confirmed by internet traffic statistics and other global data. As a result, it is becoming increasingly difficult for consumers to identify desirable media. Accordingly, there is a growing need for techniques to recommend videos or automatically classify subjects. Much effort has been made to process video. Many recent advancements in artificial neural networks have been applied to video processing in an attempt to understand each frame of the video using a convolutional neural network (CNN) [7]. Other methods used include VLAD [6] for processing time series data, recurrent neural network network (RNN) series and long short-term memory (LSTM) [4] or GRU [3] for processing time series data. The skipLSTM and skipGRU [2], which add a skip connection to the RNN network, have been proven effective ways to process time series data. The present paper uses NetVLAD and NetFV, which are known as effective methods for video processing, to find the optimal network by adjusting various hyperparameters used in the network. A single model was sought to solve the problem rather than an ensemble technique. In

* These two authors contributed equally

this process, YouTube-8M dataset was used, and a GAP rating of 0.8668 was obtained for the test set.

2 Methods and Materials

2.1 Dataset

The total number of video in YouTube-8M dataset is 6.1 million. The training set consists of 3.9 million videos. The test and validation sets are each 1.1 million. All video has an average of three labels, and each label is composed of 3,862 multi-labels. Every video is between 120 and 500 seconds in length. This paper use frame-level and audio features. The frame-level features are 1,024-dimensional vectors in which selects one frame per second in video and extracts through Inception V3 model. The audio features are extracted with 128-dimensional vectors drawn through a VGG-inspired acoustic model.

2.2 Models

This paper used NetVLAD [1] and NetFV [10], which were the most successful models used by Willow, the first place–winning team of the YouTube-8M video understanding challenge [8]. A hyperparameter was identified to match the 2nd YouTube-8M video understanding challenge limit (model size < 1 GB). NetVLAD and NetFV model uses integrated frame-level features and audio features.

2.3 Loss Function

This paper used the Huber loss function [5], which was used by SNUVL X SKT when the team earned 8th place in the YouTube-8M video understanding challenge [9]. The Huber loss combines L2 loss and L1 loss as shown in Equation 1. As the YouTube-8M dataset is substantially imbalanced by a label, the Huber loss function was used to somewhat reduce the noise.

$$L_\delta(a) = \delta^2 \left(\sqrt{1 + (a/\delta)^2} - 1 \right) \tag{1}$$

2.4 Evaluation Metric

In this paper, the global average precision (GAP) is used as an evaluation method. The GAP is calculated with the top N predictions sorted by confidence score as shown in Equation 2.

$$GAP = \sum_{i=1}^{N} p(i) \Delta r(i) \tag{2}$$

In Equation 2, $p(i)$ is the precision and $r(i)$ is the recall. In the 2nd YouTube-8M video understanding challenge, N is set to 20 and this paper is calculated accordingly also.

3 Experiments

The following experiments were performed: performance comparison of epoch, performance comparison by learning rate, performance comparison of modified dataset preprocessing.

3.1 Epoch

Of particular interest is the evaluation performed of each validation dataset for each epoch. Usually, dozens of epochs are trained in the image training process. However, that was not necessary for the YouTube-8M dataset. The validation results for each epoch are shown in Fig. 1. In the case of the YouTube-8M dataset, the training of one epoch was performed in approximately 25,000 steps when setting the batch size was 80 and used 2-GPU (total 160-batch). Although the GAP difference was slight, it was possible to observe an optimal training performance between approximately 2.5 and 3 epochs. In addition, the GAP for the training set increased in the additional training, but for the validation set decreased. Similar trends were found for various parameters of various models. In this way, it was discovered that not many epochs were needed in the training process of this dataset, thus training was completed after 2.5 epochs.

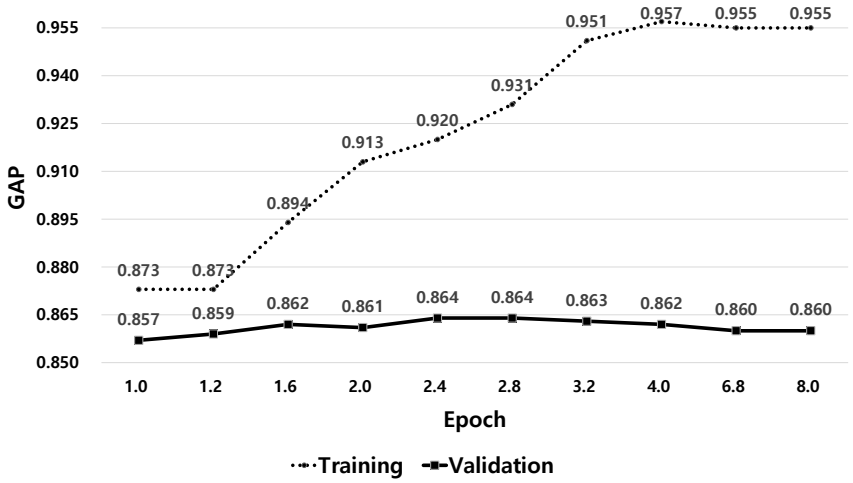

Fig. 1. GAP per epoch curve. The dotted line represents GAP per epoch in the training set; the solid line is GAP per epoch in the validation dataset. As the epoch increases, the training GAP curve also increases. However, the validation GAP curve shows a trend of declining after about 2.5 epochs.

3.2 Learning Rate

The epoch experiment revealed that overfitting of the training set occurs when the model continues to train more than 3 epochs. This paper resolves this problem by adjusting the learning rate to be more effective. In the early part of training, it is good to provide a relatively high learning rate to ensure quick training. Conversely, at the end of the training, the learning rate should be decreased. Thus, the experiment began with a high learning rate, which was set to diminish over time. The learning rate decay per epoch was set to 1/10 of the baseline. It also increased the initial learning rate by 10 times that of the baseline. These ways have helped reduce overfitting. The learning rate decay was 0.8, for the purpose of keeping the learning rate of the two methods similar when the first epoch is complete. So that the learning rate was greater than the baseline even if the training data is in the latter half. This ensured training about these data. The learning rate change at this method is shown in Fig. 2. The resulting performance improvement is shown in Table 1.

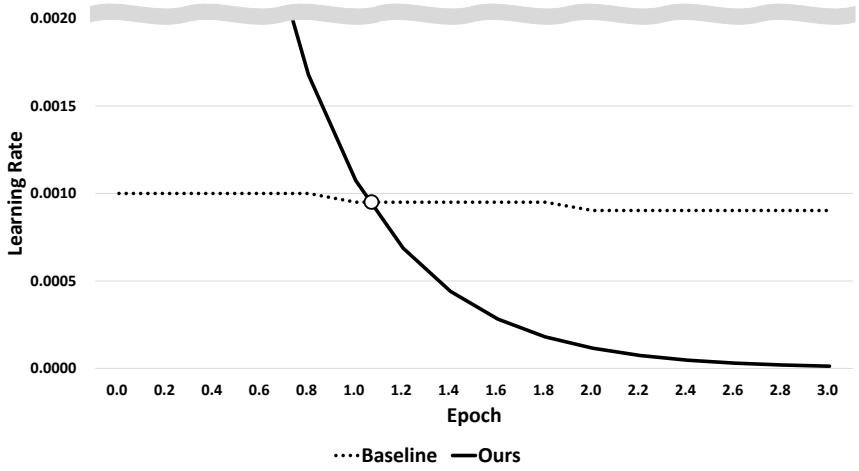

Fig. 2. Comparison of baseline and experimental learning rates. The initial learning rate of the present method was set at 10 times that of the baseline and the decay per epoch at 1/10 the baseline. The learning rate decay was modified so that the learning rate when the first epoch passed was similar to that of the baseline.

3.3 Data Preprocessing

There was an attempt to improve performance through dataset modifications. First, the imbalance of the dataset was identified and addressed. Second, the false values of the results obtained were analyzed by validating the data trained with the default training set.

Table 1. Hyperparameters and its GAP score. Hyperparameters were modified. GAP increased by 0.002.

Hyperparameter	Baseline	Ours
Initial learning rate	0.001	0.01
Learning rate decay	0.95	0.80
Learning rate decay per epoch	1.0	0.1
GAP	**0.864**	**0.866**

Overfit to Non-dominant Pattern Fig. 3 illustrates that the top 900 labels in the training set accounted for 89% of the multi-label video data, or 10,445,267 of the 11,711,620 total labels for video data. The remaining 2,962 labels represent only 10% of 1,266,353 individuals. To solve this data imbalance, a small training set was constructed with a label index > 977. In this small training set, one epoch of training is performed at 7,300 steps with 2-GPU and each 80-batch (160-batch in total). This small training set was used in two ways. The first method was to train the small training set when the train GAP converged to 1.0 and retrain it as the existing default training set. However, this performance was lower than that of the existing GAP of 0.86 (see Fig. 4). In the second method, 2.5 epochs were trained with the default training set and retrained with a small training set (see Fig. 5). But, performance dropped when trained with a small training set.

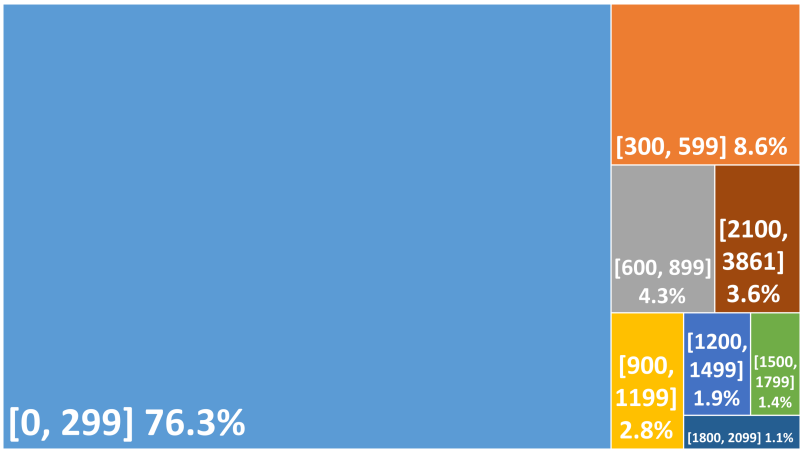

Fig. 3. Visualization of training set imbalance. The top 300 labels represent 76% of the total multi-labels. Expanding the range to the top 900 labels takes up 89% of the total.

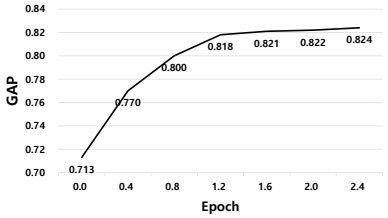

Fig. 4. GAP curve about validation set: Training with a small training set and retraining with the default training set. 0.0 Epoch means when overfitted with the small training set.

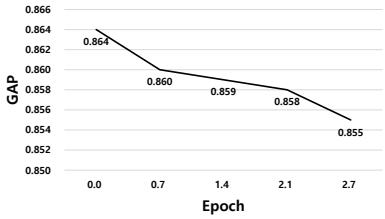

Fig. 5. GAP curve about validation set: Training with the default training set and retraining with the small training set. 0.0 Epoch is when 2.5 epochs were trained with the default training set.

Analysis of Validation Results and Additional Experiment The correct answer was compared to the top 20 prediction results of the model. The model validation GAP is 0.86. The analysis showed that 117,410 in a total of 1,112,357 validation set did not include some or all of the correct answers in the top 20 predictions. Of these, 21,828 were single-label, and 55,470 had more than 4 labels. Those with 1 label involved a unique feature of the video label, and those with 4 or more labels had overlapping label features and did not train well. So, training data was selected with only 1 or 4-plus labels. These data were added to the training data by tripling them from other data. In all, about 8,500,000 large training set was created and trained. Unfortunately, there was no significant performance improvement; with a difference of only about 0.0001 according to the GAP, no performance improvement was found through dataset preprocessing. A clearer interpretation method is needed.

3.4 Final Submission Model

An optimal hyperparameter for the NetVLAD and NetFV models was found through the above methods. The results of the test set are shown in Table 2.

Table 2. The performance (GAP) of each model. Optimal cluster size, hidden size and GAP are shown.

Model	Cluster size	Hidden size	GAP
NetVLAD	192	1,200	0.86668
NetFV	120	1,024	0.86633
Result			**0.86668**

In the end, a GAP score of 0.8668 was obtained at the 2nd YouTube-8M video understanding challenge.

4 Conclusions

This paper used video classification of the YouTube-8M dataset, applying the NetVLAD and NetFV models with reference to previous research data and using the Huber loss function. Experimental verification is effective for improving performance by adjusting the training epoch, learning rate, and training set. Unlike in conventional training for classification problem, the performance of 2.5 epochs is found to be optimal, as the training set is sufficiently large. The learning rate was also adjusted for optimal training. Even though no performance improvement was found, an attempt was made to train with the set that emphasized frequently-wrong patterns.

Acknowledgement This work was partly supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (2017-0-01772, Development of QA system for video story understanding to pass Video Turing Test), Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (2017-0-01781, Data Collection and Automatic Tuning System Development for the Video Understanding), and Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (No.2017-0-00271, Development of Archive Solution and Content Management Platform)

References

1. Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: Netvlad: Cnn architecture for weakly supervised place recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5297–5307 (2016)
2. Campos Camunez, V., Jou, B., Giró Nieto, X., Torres Viñals, J., Chang, S.F.: Skip rnn: learning to skip state updates in recurrent neural networks. In: Sixth International Conference on Learning Representations: Monday April 30-Thursday May 03, 2018, Vancouver Convention Center, Vancouver:[proceedings]. pp. 1–17 (2018)
3. Cho, K., Van Merriënboer, B., Bahdanau, D., Bengio, Y.: On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259 (2014)
4. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
5. Huber, P.J.: Robust statistics. In: International Encyclopedia of Statistical Science, pp. 1248–1251. Springer (2011)
6. Jégou, H., Douze, M., Schmid, C., Pérez, P.: Aggregating local descriptors into a compact image representation. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. pp. 3304–3311. IEEE (2010)

7. Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882 (2014)
8. Miech, A., Laptev, I., Sivic, J.: Learnable pooling with Context Gating for video classification. ArXiv e-prints (2017)
9. Na, S., Yu, Y., Lee, S., Kim, J., Kim, G.: Encoding Video and Label Priors for Multi-label Video Classification on YouTube-8M dataset. ArXiv e-prints (2017)
10. Perronnin, F., Dance, C.: Fisher kernels on visual vocabularies for image categorization. In: 2007 IEEE conference on computer vision and pattern recognition. pp. 1–8. IEEE (2007)