CS230

Multi-label Video Classification

Wei-Ting Hsu
hsuwt@stanford.edu
Stanford University

Mu-Heng Yang
mhyang@stanford.edu
Stanford University

Abstract

We proposed 10 models to large scale video classification using the recently published YouTube-8M dataset, which contains more than 8 million videos labeled with 4800 classes. Among 10 proposed models, we find best performance on Stacked Bi-LSTM ensemble with MoE using both visual and audio features. After hyperparameters tuning, we achieve a Hit@1 of 85.7%, a significant improvement over the 64.5% for the best model Google trained in their initial analysis.

1 Introduction

The capability to understand videos with machine would be valuable across social media. To open the door to this problem, we think classifications of videos can be the first step towards video understanding. Inspired by the great success of CNNs on image recognition tasks and RNNs on sequence labeling, video classification presents a great research area to experiments with various kind of hybrid deep learning models. Previous. Previous research [14] was limited due to the expensive nature of having labeled video data. However, the recently release large-scale data, YouTube-8M [2] by Google is of the scale comparable to ImageNet [5], and thereby stimulates many interesting approaches to this task. We will investigate the problem and improve upon techniques that have shown to work.

2 Related work

2.1 YouTube-8M: A Large-Scale Video Classification Benchmark

This paper introduces the motivation of releasing the YouTube-8M datasets, which is to advance the field of video-understanding. It summarizes the way the tackle the challenges of collecting huge video dataset and pre-processing the videos into features. They also provided simple benchmark and demonstrates the ability to generalize to other domain-specific videos data like Sports-1M [9] and ActivityNet [6].

2.2 Learnable pooling with Context Gating for video classification

This paper [10] tackle the problem of learning from sequential feature vectors that are complicated in natures, in this case, sequence of video images. It experimented with several pooling techniques such as Soft Bag-of-words, Fisher Vectors [12], NetVLAD [3], which out performs GRU [4] and LSTM [8], which all have learnable parameters. The paper also introduces a new mechanism called Context Gating, which is shown to be beneficial for the trainable versions of Bag-of-words, VLAD and Fisher Vectors.

3 Dataset and Features

The dataset we will be using: YouTube-8M [2] contains 8,264,650 videos data. We split videos into 3 partitions, Train : Validate : Test, with ratios 70% : 20% : 10%. These videos are extracted randomly from YouTube and are filtered to have visually recognizable entities. The dataset contains 4716 entities, or label classes, where each entity contains at least 200 videos. On average each video data is associated with 3.4 entities. The data is in the format of feature vectors extracted in sequential video frames. Frame-level features include visual and audio ones. The visual features were extracted using Inception-V3 [13] trained on ImageNet [5]. The audio features were extracted using a VGG-like acoustic model [7]. Both features were PCA-ed. The combined frame-level are 2TB in size. Video-level features vector is also provided which is an aggregation of frame-level features. Both types of features have shown competitive results [11] and both deserves experimentations with advanced models.

4 Baseline

4.1 Logistic Regression

Logistic regression is used on both video-level and frames-level features. For video-level, the 1024 features of a video are linearly projected to a output space, then pass through a softmax layer to convert logits into probabilities. For frame-level, the label of the entire video is used as the label of each individual frames. Twenty frames for each video are randomly sampled and are fitted into a logistic regression model. In prediction time, probabilities of each class are aggregated by averaging across all frames within the video.

4.2 Bag of Frames

Similar to Bag of Words where words within a sentence is trained across a neural network with shared parameters. The BoF model regard frames as constituting components that can be aggregated to represent the label class of a video. In this model, a subset of frames are randomly selected and trained across a deep neural net with shared weights into a larger projection vectors. All projection vectors are average-pooled then propagate into a softmax classifier.

4.3 LSTM

Using RNN on top of the sequential frames of videos, the LSTM model is able to capture the temporal information. As the model name suggests, LSTM cells are used in the construction to preserve long-term memory since the videos can have up to 500 frames. Two hidden layers of 1024 units each are adopted in this baseline.

4.4 Mixture of Experts

Mixture of Experts is a dense neural network architecture that consists of several parts: n-number of experts network, and one gating network. Each expert is a fully connected network where the input is a feature of $\mathbb{R}^{1024}$, and output a vector of $\mathbb{R}^{k}$. The gating network predicts the probability of choice amongst these network and also takes the input of $\mathbb{R}^{1024}$, and output a vector of $\mathbb{R}^{n}$. The final probability prediction is $\sum_{i=1}^{k} p(i|x)E_i(x)$ where $p(i|x)$ is the output of gating network for i^{th} expert and E_i is the output of i^{th} expert network.

5 Proposed Model

Our proposed strategy comes in two part: a) develop a robust classification model on video-level data, and b) aggregate frames-level data into informative representations of videos. For each task, we experimented with various architectures, and select the best model that excels in each and combine them to breed our final model of video tagging.

5.1 Video-Level Classifiers

5.1.1 Deep Mixture of Experts

Similar to MoE, Deep MoE learns a gating network and several expert network, except that each expert network is deepened with more hidden layers. The gating network is used to assign weights of each expert networks as before. We tuned this model to have 3 hidden layers with 4096 hidden units each.

5.1.2 Residual Network

As an attempt to experiment with deep network without increasing too much of the difficulty of training, Residual Network is implemented. It is a deep fully-connected neural network with skipped connections at every two layers, which ameliorate with gradient diminishing problem. The final output layer uses softmax to product probability predictions for each class.

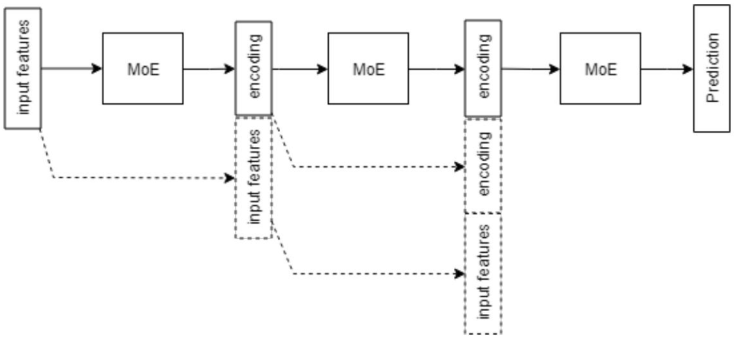

Figure 1: Chain Mixture of Experts

5.1.3 Chain Mixture of Experts

Inspired by the potential of MoE, we want to exploit the ensemble nature of these mixtures by training several MoE at once. With typical MoE, each experts is trained to make predictions on the k classes. Whereas in Chain MoE, each MoE is trained to learn an encoding of its input features to a space of e dimensions, The e-dimensional vector is then used as the input to the next MoE, along with a copy of the video-level features and encodings from all previous stages of MoE. This is illustrated in Fig. 1. The number of chain units can be arbitrary, but with higher length, the model becomes increasingly difficult to train without prominent performance gain, as loss is hard to propagate through early MoE units. We compare results of Chain MoE with length 2 and 3 in the subsequent sections. The intermediate encoding vector is tuned to have a dimension of 128.

5.1.4 Deep Chain Mixture of Experts

As a variant to Chain Mixture of Experts, we also experimented with additional fully-connected layer between each chain units. In this model, instead of making our chain units to directly project encoding of 128-dimensional vector, they project encoding equals to the dimension of vocab size (i.e number of classes), then have the fully-connected layer to reduce dimension to 128. In our experiment, we select the length of Chain MoE to be 3.

5.2 Frames Aggregation

5.2.1 LSTM/GRU

We adopted the baseline LSTM and introduces GRU variant for faster training, and experimented with different number of layers and hidden units.

5.2.2 Stacked Bidirectional LSTM/GRU

Even though video class is defined primarily based on forward sequence of video frames, bidirectional RNN may help in a sense that images at later frames can help understanding information at current frames. We stack two layers of RNN: first layer being a bidirectional, second layer being a unidirectional. The outputs of each time steps in the first layer of bidirectional RNN are concatenated together and used as the time input of the second layer RNN. The last hidden state of the unidirectional RNN is then used as our final aggregated feature. The architecture is illustrated in Fig. 2.

5.2.3 Attention

Inspired by attention model in traditional encoder-decoder scheme, we thought that each feature may have different importance at different time step. Therefore, we proposed an attention model where frames features are first unrolled through LSTM/GRU, then the outputs of k dimensions at all max_t time-steps are feed through a fully-connected layer and condensed into a k by 1 vector, which is used as our final aggregated features. Note that before feeding into fully-connected layer, the outputs are masked according to the length of

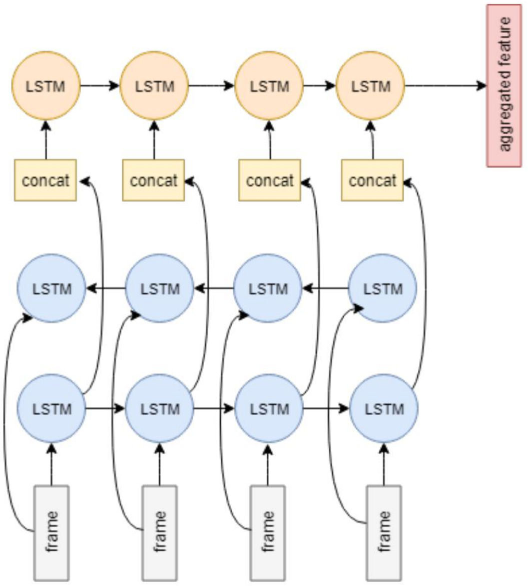

Figure 2: Stacked Bidirectional LSTM

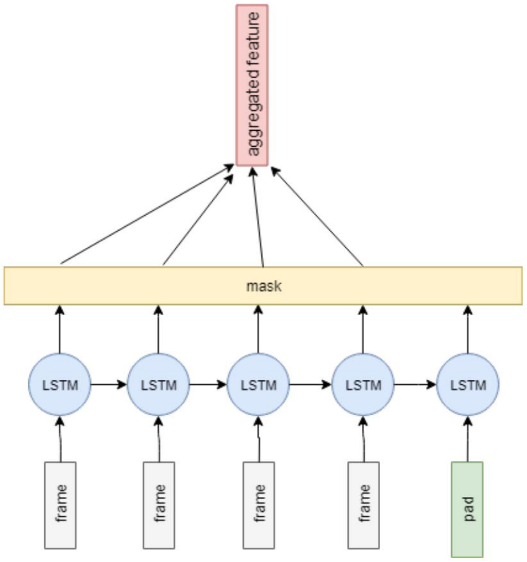

Figure 3: LSTM with Attention

individual videos, and that time-steps that are longer than the video length are not considered. The architecture is illustrated in Fig. 3. The RNN is tuned to have 2 layers of 1024 cells each.

5.2.4 Weighted Frames

Instead of learning the importance across each time-step, we could also adopt attention principle to learn the importance across the k-dimensional RNN outputs. In the Weighted Frames model, for each RNN outputs at each time-step, we apply an attention fully-connected layer and condense the k-dimension into 1-dimension. This is can be interpreted as the weight of each time-step. The weights are normalized and multiplied by the frame-level features and summed to produce the final aggregated features. Same as Attention models, weights are masked on/off along the time axis to ignore padded time-steps. The RNN is tuned to have 2 layers of 1024 cells each.

6 Experiments/Results/Discussion

6.1 Global Hyperparameters

Our models are developed using TensorFlow [1]. After hyperparameter tuning, we chose 0.01, 0.0002 as learning rate for video-level and frame-level models respectively. As there are more noise in frame-level features, learning rate needs to be much lower to make models converge smoothly. Batch size is 1024, 128 for video-level and frame-level models respectively due to limited memory for frame-level model. All models are trained for 5 epochs with 0.95 learning rate decay and 0.5 dropout rate using Adam optimizer. In all frames-level aggregation method, we chose the number of cells in RNN to be 1024 with 2 layers. As a reference, it takes over one day to train a two-layer LSTM model because there are 1.7 TB features for training.

6.2 Evaluation Metrics

We evaulate avg Hit1, avg PERR, MAP, and GAP in our project. For each video, we will predict a list of confidence scores for 4800 labels. Hit1 measures the percentage of test data that contains at least one of ground truth in its top one predictions. PERR (precision at equal recall rate) measures the precision of each test data in the top x predictions, where x is the number of ground truth label associated with the video, and then taking the mean of precision across all samples. GAP (gloval average precision) represents the area under the precision / recall curve. If a submission has N predictions (label/confidence pairs) sorted by its confidence score, then the Global Average Precision is computed as: $GAP = \sum_{i=1}^{N} p(i)r(i)$ where N is the number of labels multiplied by number of videos in prediction, p(i) is the precision, and r(i) is the recall. MAP (Mean Average Precision) is similar to GAP, the only difference is that N is the number of labels, then we take mean of AP over number of videos in prediction.

	Model	Avg Hit1	Avg PERR	MAP	GAP
Baseline	Logistic Regression	0.825 / 0.788	0.691 / 0.646	0.399 / 0.375	0.759 / 0.707
	MoE	0.839 / 0.805	0.709 / 0.668	0.415 / 0.396	0.782 / 0.742
Proposed	Deep MoE	0.826 / 0.774	0.686 / 0.626	0.273 / 0.200	0.758 / 0.696
	Deep Chain MoE	**0.852** / 0.826	**0.725** / 0.693	**0.427** / 0.411	**0.799** / 0.768
	Chain MoE	0.848 / 0.819	0.720 / 0.684	0.423 / 0.406	0.795 / 0.761
	Chain MoE3	0.850 / 0.821	0.722 / 0.688	0.424 / 0.406	0.797 / 0.764
	ResNN	0.106 / 0.175	0.001 / 0.048	0.000 / 0.000	0.000 / 0.003

Table 1: Evaluation on validation set using video-level models with/without audio features

	Model	Avg Hit1	Avg PERR	MAP	GAP
Baseline	Logistic Regression	0.785 / 0.738	0.633 / 0.580	0.167 / 0.139	0.677 / 0.614
	Deep BoF	0.840 / 0.811	0.705 / 0.670	0.332 / 0.305	0.777 / 0.742
	LSTM	0.856 / 0.835	0.728 / 0.702	0.378 / 0.370	0.802 / 0.778
Proposed	GRU	0.856 / 0.830	**0.729** / 0.698	**0.408** / 0.389	0.803 / 0.773
	Stacked Bi-LSTM	**0.857** / 0.833	**0.729** / 0.700	0.391 / 0.373	**0.804** / 0.777
	Stacked Bi-GRU	0.856 / 0.830	0.728 / 0.698	0.407 / 0.389	0.803 / 0.774
	LSTM Attention	0.853 / 0.829	0.722 / 0.694	0.352 / 0.337	0.796 / 0.768
	Weighted Frames	0.788 / 0.760	0.638 / 0.606	0.214 / 0.191	0.702 / 0.673

Table 2: Evaluation on validation set using frame-level models with/without audio features

6.3 Results & Discussion

6.3.1 Video-level Classifier

The results of various video-level models are shown in Table 1. All models are trained on the best hyperparameters we have found. Note that by tuning the hyperparameters, we were able to surpass the benchmark results from Google [2] by huge margin. The best model is Deep Chain MoE across all metrics, and Chain MoE3 comes in second with little difference. We decide to select Chain MoE3 in our final model because of its better computational efficiency with minimal performance sacrifice. All models achieves higher score with audio features, which is expected since more information is provided with audio.

6.3.2 Frames Aggregation

In evaluating the results of frames aggregation, we feed the aggregated features to the baseline MoE classifier, because MoE trains relatively fast and performs better than Logistic Regression baseline. The results of all frames aggregation with MoE classifier are shown in Table 2. We find that Bi-LSTM and GRU performs the best, with other LSTM, Bi-GRU all comes very close. Note that even though LSTM attention performs slightly worse than the best models, it actually excels when we experimented without dropout with all else equals. This mean dropout regularization actually reduces over-fitting more prominently on RNN models without attention layer, or that the fully-connected layer in LSTM attention has some effect of regularization. We again see all models perform better with audio.

6.3.3 Video-Level Models on Aggregations of Frames

We combined Stacked Bi-LSTM Frames Aggregation model with Chain MoE3 video-level classifier and find the results to be identical with Stacked Bi-LSTM without chaining: Hit@1 of 0.857. However, this result has only ran for 2 epochs as opposed to 5 epochs without chaining due to computational resources and limited time we have. We shall expect the result to be slightly better with more epochs.

7 Conclusion/Future Work

We found that using both visual and audio features can significantly increase all metrics from 2% to 4%. Proposed Chain MoE is 1.3% better than baseline MoE model. Our LSTM attention without dropout also outperform baseline LSTM model.

Due to the large amount of computational resources and time necessary to run each experiment (approximately a day), we were not able to run more extensive investigation of various affects of hyper-parameters or develop more sophisticated model. In the future, more complex ensemble method, like attention weighted stacking, for multiple video/frame-level models will be explored.

Contributions

Wei-Ting Hsu: Model development, write-up, poster

Mu-Heng Yang: Experimentation, data preparation, model tuning, write-up

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

[2] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016.

[3] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5297–5307, 2016.

[4] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[6] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 961–970. IEEE, 2015.

[7] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. Cnn architectures for large-scale audio classification. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 131–135. IEEE, 2017.

[8] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(9):1735–1780, November 1997.

[9] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[10] Antoine Miech, Ivan Laptev, and Josef Sivic. Learnable pooling with context gating for video classification. *CoRR*, abs/1706.06905, 2017.

[11] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 4694–4702. IEEE, 2015.

[12] Florent Perronnin, Yan Liu, Jorge Sánchez, and Hervé Poirier. Large-scale image retrieval with compressed fisher vectors. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3384–3391. IEEE, 2010.

[13] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

[14] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence-video to text. In *Proceedings of the IEEE international conference on computer vision*, pages 4534–4542, 2015.