Semantic Video Classification

Xiulong Liu, Tianyi Zhang

Introduction

Semantic video classification is providing semantic tags for videos, where each video can be associated with multiple categories. For example, a video can be categorized as sport, and further to be basketball. In our project, we aim at predicting the probability of each video belongs to each category and choosing the top 5 as our labels for the result. Our data comes from YouTube8M dataset, including 3862 total categories with 6.1million videos. And we treat the problem as multiple binary classification problem.

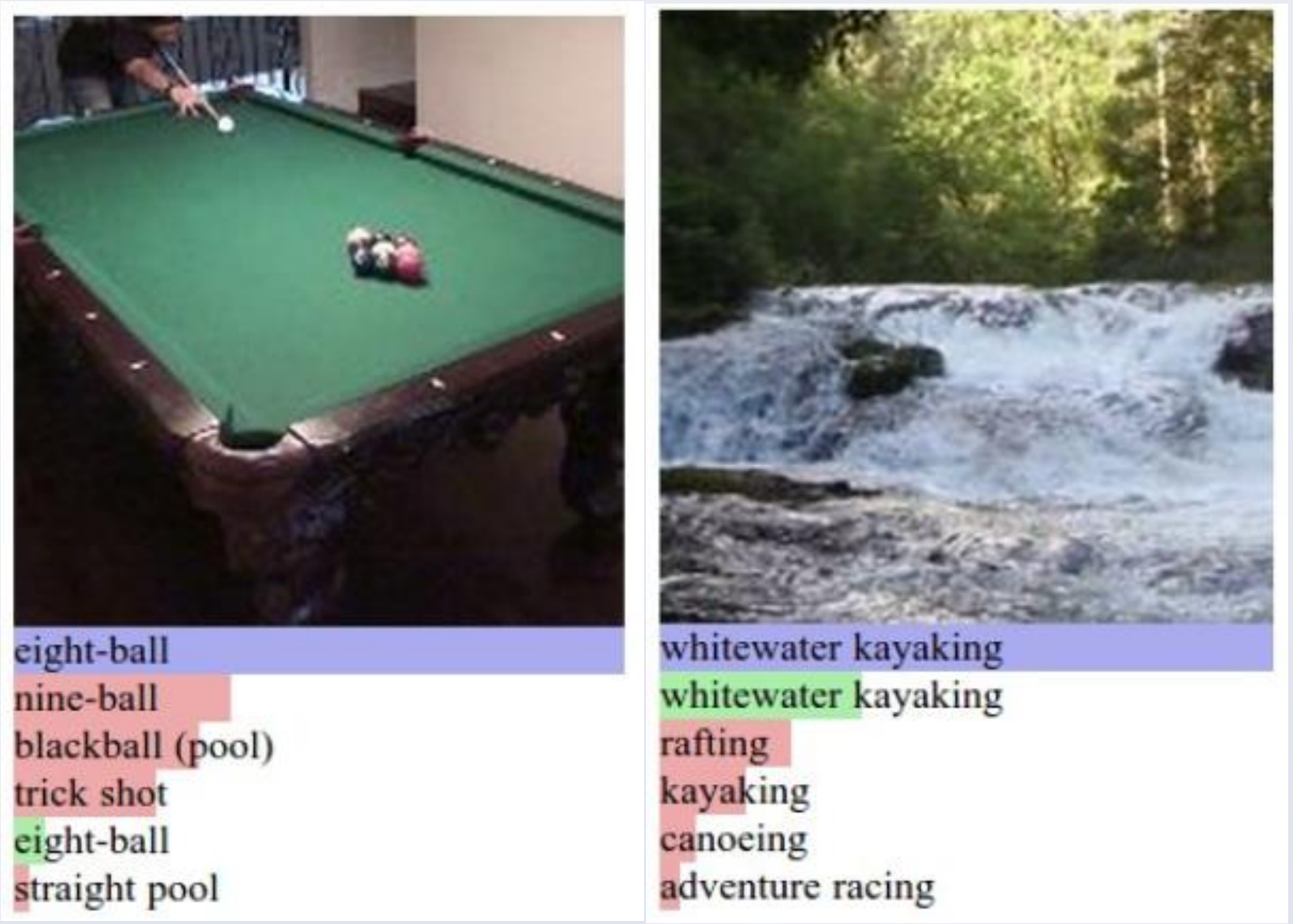


Figure 1. Examples from dataset

Dataset Structure

Youtube8M dataset contains over 6 million videos with 3862 labels. The average number of label is 3 per video. Two types of features are available, frame-level and video-level. All features are obtained from a GoogLeNet pre-trained on ImageNet. For our semantic video classification need, we uses the frame-level feature. To obtain frame-level feature, the original video is down sampled into 300 frames. For each frame, a 1024-dimension feature is generated. Along with video id and label, these three types of content are packed into a tfrecord file. Due to the massive among of data, we only used 10% of the total 6 million video to train our network.

Models

We uses LSTM model to treat the input data as a sequence to obtain the feature in temporal feature. LSTM is derived from RNN model. RNN contain multiple connected cells. Each cell can have its own input and output. The weight of each cell is passed to the next cell to preserve the features in sequence input. LSTM is designed for the network to remember the relatively important input in each cell and filter out the input that does not contribute much to the output. LSTM uses sigmoid to act as gates for the information to pass selectively. All gates in a LSTM cell control the state of this cell. The output of each cell is determined by the state. We use two layers of stacked LSTM, each layer contains 512 cells. For a 300-frame input feature, each cell takes a frame as input and output labels. Only the last output is considered result label and output in other cells are used for loss calculation and back propagation. We use sigmoid for each individual label to output a probability. The loss function is implemented is sigmoid cross entropy to estimate the loss across 3862 labels. For performance comparation, we also implement a logistic model that has one fully connected layer with sigmoid function.

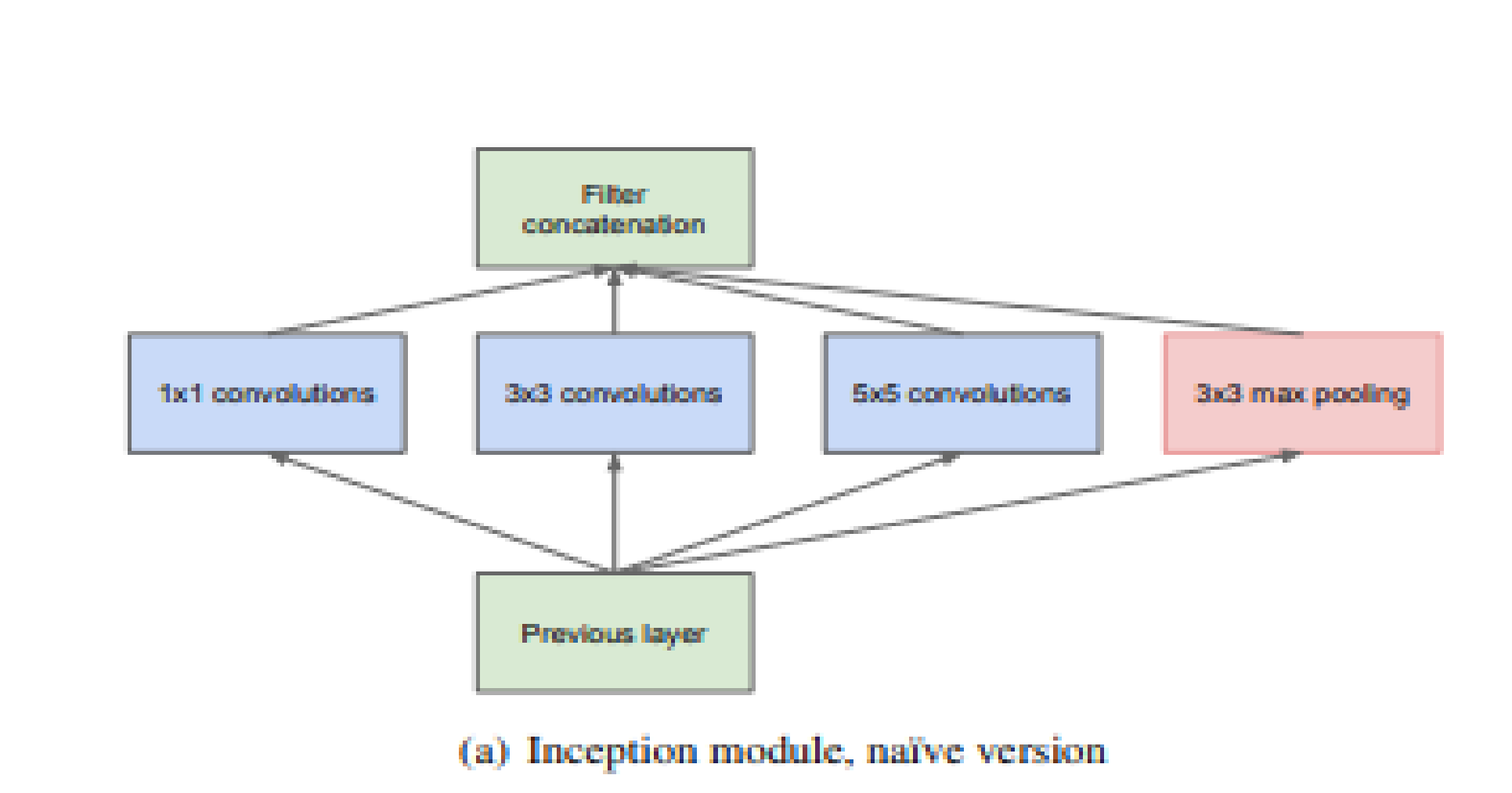


Figure 2. Inception module used in GoogLeNet

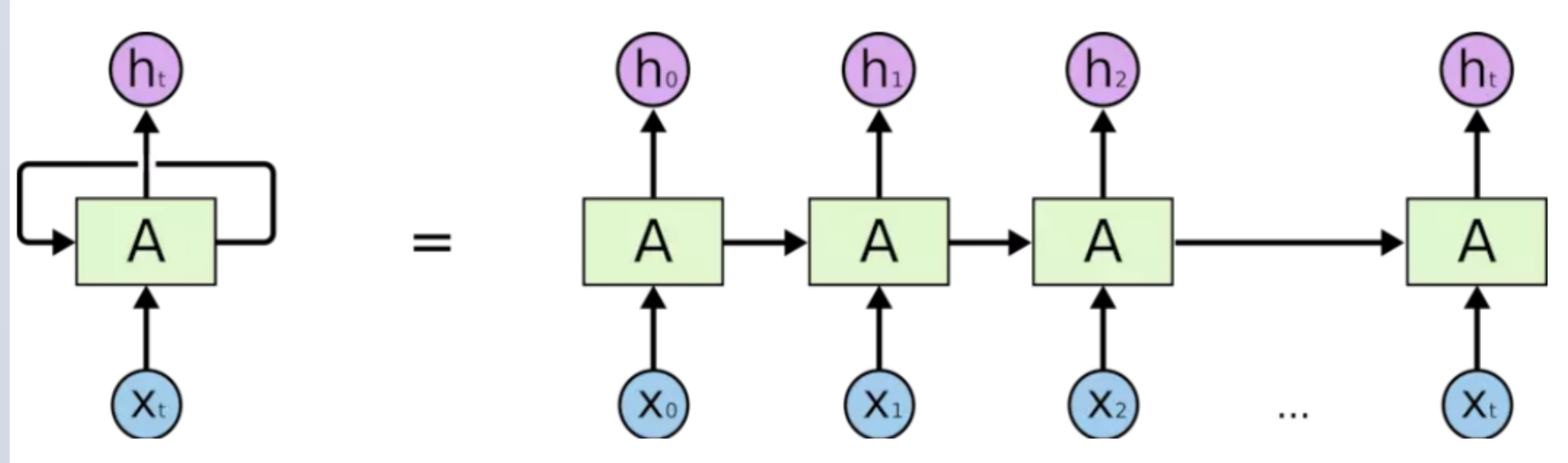


Figure 3. RNN model

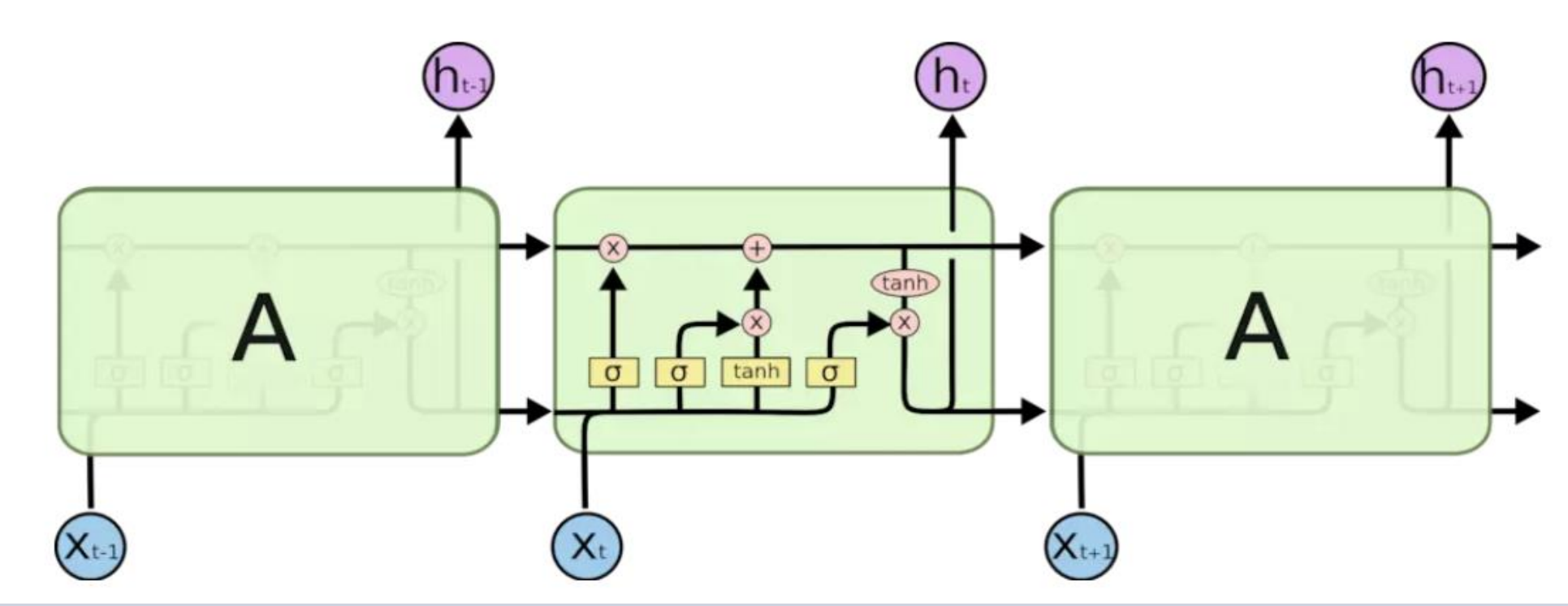
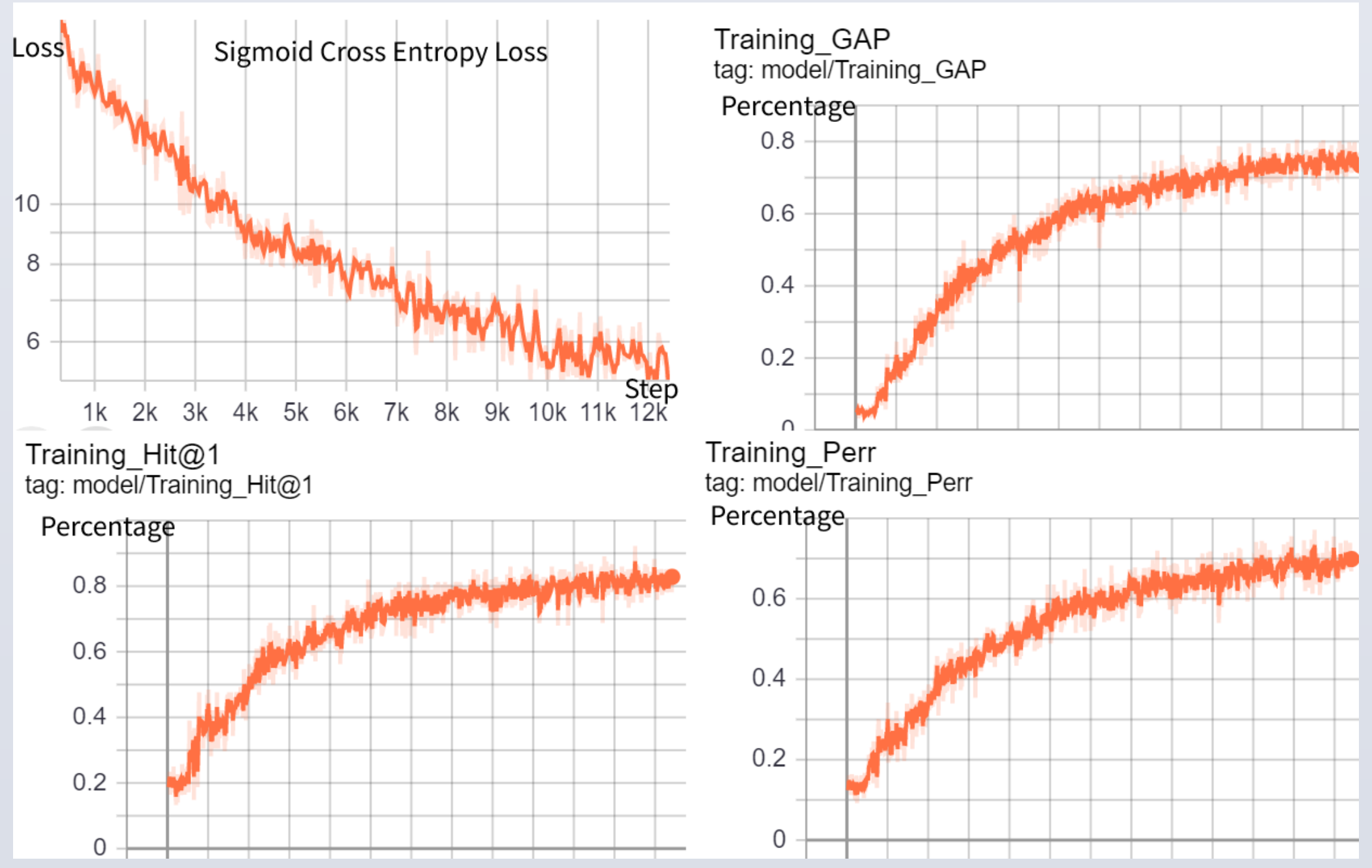


Figure 4. LSTM cells

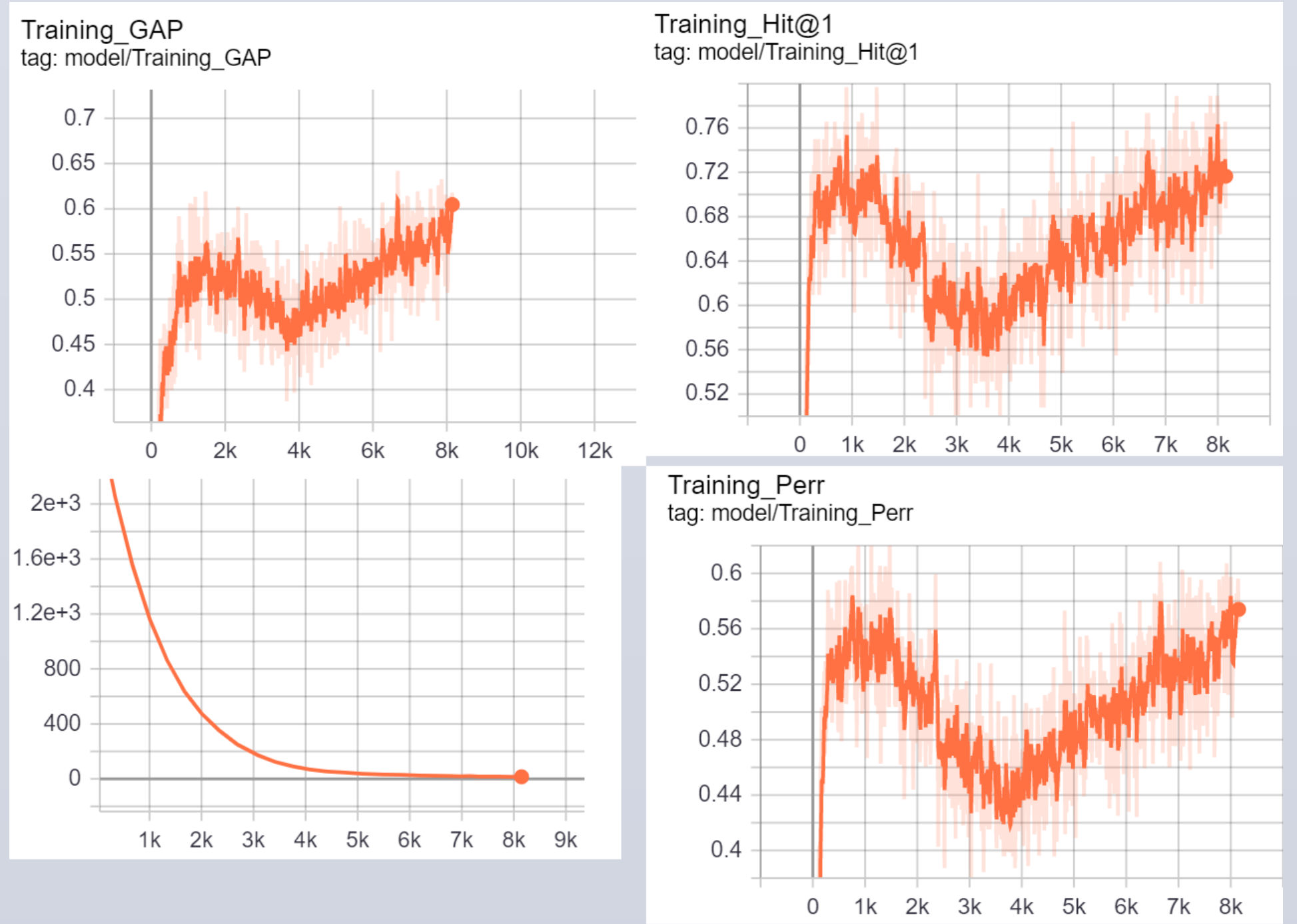
Experiments and Results

We use 600000 videos to train our LSTM model and 10000 videos to test. Since the dataset is massive, we only train our model for 5 epochs with the learning rate of 0.001, batch size of 128, 0.05 of learning rate decay every 400000 examples and Adam optimizer. The total time used for training 5 epoch is over 12 hours. We uses three different indices to evaluate the performance of the model[2]. GAP is the average precision on the top 20 labels per example. Hit@1 is the probability of the top 1 label being the ground truth label. PERR is the annotation precision when the same number of labels per video are retrieved as there are in the ground-truth[3]. Our LSTM model result is shown in figure 4. We also train the logistic model on the same dataset. The result is shown in figure 5.



```
epoch/eval number 12238 | Avg_Hit@1: 0.810 | Avg_PERR: 0.683 | MAP: 0.212 | GAP: 0.736
```

Figure 4. Result of LSTM model

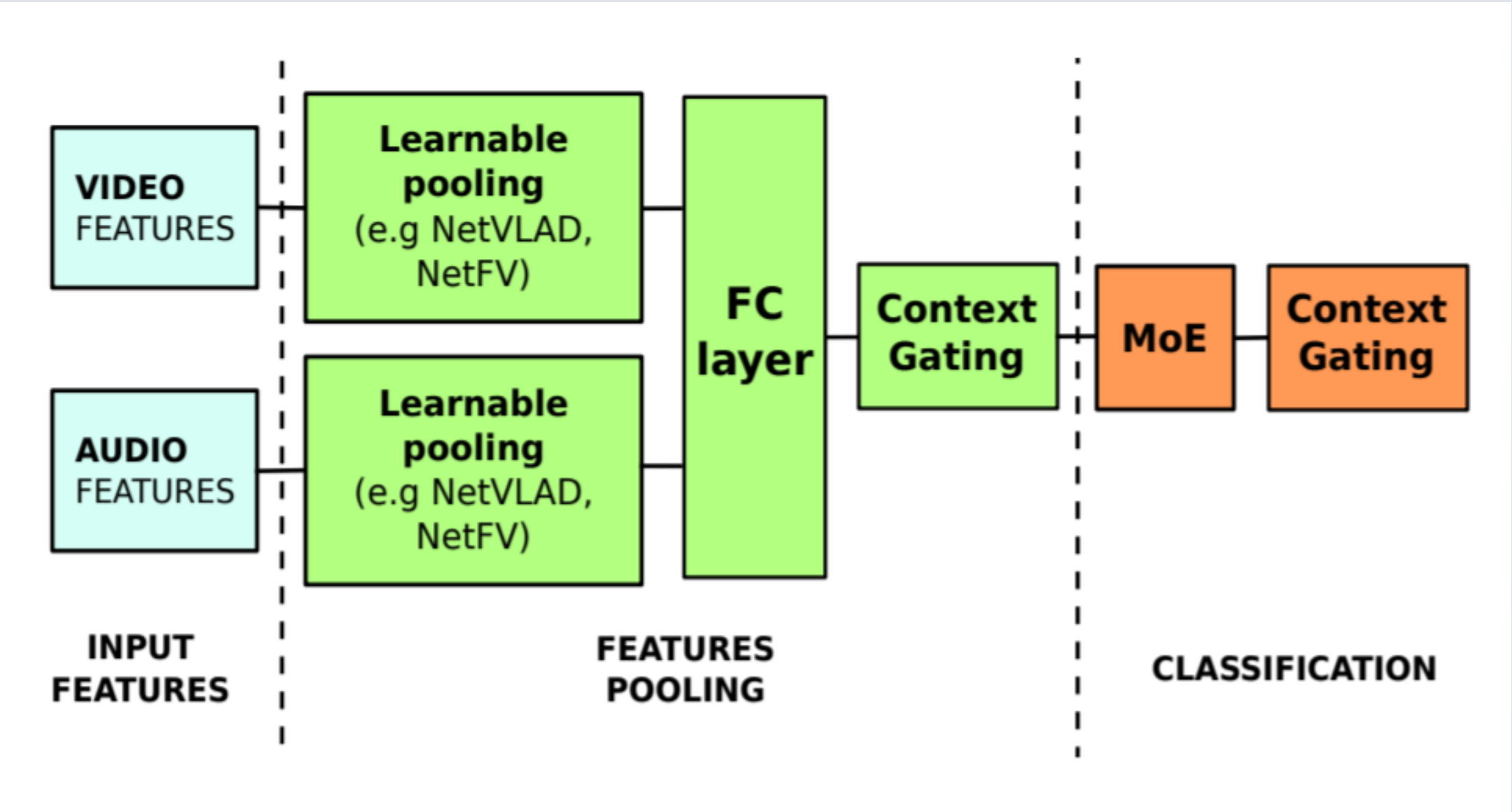


```
I0610 21:35:35.633991 139765123482176 eval.py:250] Done with batched inference. Now calculating global performance metrics.
I0610 21:35:38.741825 139765123482176 eval.py:262] epoch/eval number 8010 | Avg_Hit@1: 0.710 | Avg_PERR: 0.557 | MAP: 0.077 | GAP: 0.577 | Avg_Loss: 15.61731
```

Figure 5. Result of Logistic model

Conclusion

LSTM model and logistic model both achieve relatively good result. LSTM model overperforms logistic model by 16% in GAP on test dataset. On one hand, the logistic model is hard to retrieve temporal information, while its result is still good. This may indicate that the feature in every frame is more important than the temporal information. On the other hand, the increase of performance for LSTM model indicate that the temporal information has certain contribution to the result. The champion model of the ouTube-8M Video Understanding Challenge on Kaggle focus more on the video features rather than the temporal information. The model structure is shown in figure 6. The result GAP on YouTube8M dataset is 84.97 while our LSTM model is 73.6. A context gating unit is introduced in this model and it replaces LSTM to extract the context information.


VIDEO FEATURES

AUDIO FEATURES

Learnable pooling (e.g NetVLAD, NetFV)

Learnable pooling (e.g NetVLAD, NetFV)

FC layer

Context Gating

MoE

Context Gating

INPUT FEATURES

FEATURES POOLING

CLASSIFICATION

Figure 6. State-of-the-art model

References

[1] Szegedy, Christian, et al. "Going deeper with convolutions." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

[2] Abu-El-Haija, Sami, et al. "Youtube-8m: A large-scale video classification benchmark." arXiv preprint arXiv:1609.08675(2016).

[3] Kim, Hyun Sik, and Ryan Wong. "Google Cloud and YouTube-8M Video Understanding Challenge.“

[4] Miech, Antoine, Ivan Laptev, and Josef Sivic. "Learnable pooling with context gating for video classification." *arXiv preprint arXiv:1706.06905* (2017).