The 2nd YouTube-8M Video Understanding Challenge (youtube8m)

Vladimir Aliev
Gleb Sterkin
Pavel Ostyakov
Elizaveta Logacheva
Roman Suvorov
Oleg Khomenko

SAMSUNG AI Center
- Moscow

(#4)

SAMSUNG AI Center
- Moscow

Problem statement

Problem

Multilabel classification problem with **avg. labels per video ~ 3.0** out of **3862 classes**;
Labels are **automatically generated** with the YouTube video annotation system;
Final model should be TF Graph and meet 1Gb size requirement.

Data

- Updated youtube8m dataset with **improved** quality **machine-generated labels**,
- and **reduced size** video dataset;
- Hidden representation produced by Deep CNN pretrained on the ImageNet dataset; for both **audio spectrogram and video frames taken at rate of 1Hz**;
- The dataset also contains **aggregated video-level features extracted as averaged frame-level features**;
- 1024 video features; 128 audio features;
- Frame-level train: 1.3 Tb; Frame-level test: 268 Gb;
- Video-level train: 12 Gb; Video-level test: 2.5 Gb;
- Data was converted from tf.records to np.ndarray.

Evaluation

Evaluation metric — GAP@20

The GAP metric takes the predicted labels with the highest k=20 confidence scores for each video, treats each prediction as an individual data point in a long list of global predictions sorted by their confidence scores. The list is then be evaluated with Average Precision across all of the predictions and all the videos:

$$AP = \sum_{i=0}^{N} p(i)\Delta r(i)$$

where *N* = 20 × number if videos, *p(i)* is the precision, and *r(i)* is the recall given the first i predictions.

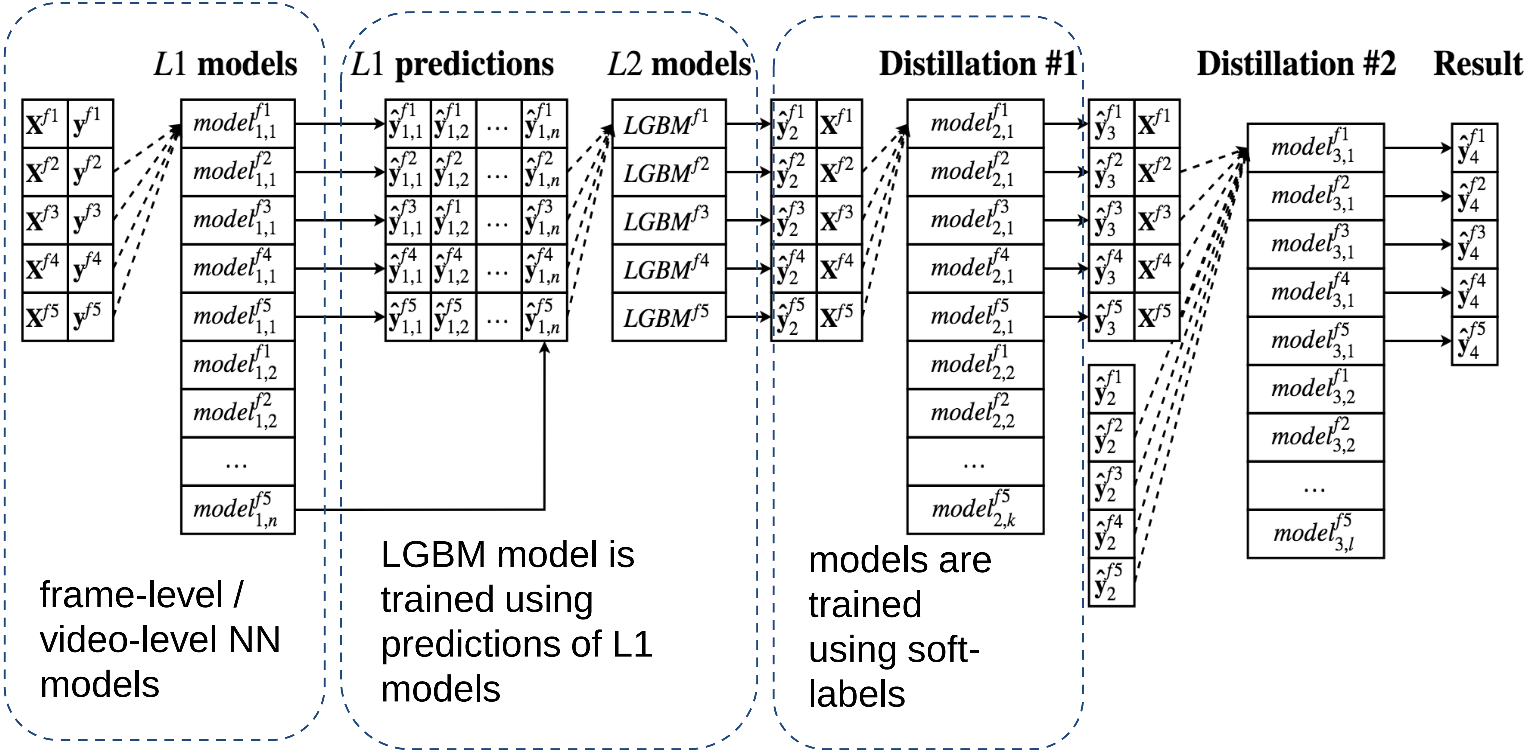
General approach

Our team sticked to the following approach:

- Train various first-level models;
- Train an ensemble on predicted labels using LightGBM;
- Extract out-of-fold predictions from the ensemble;
- Train several models using soft-labels;
- Finally, train second-level NN.

Loss. Binary cross-entropy was selected as main loss function, although other options were also tried (soft ranking loss, hinge ranking loss). Reweighting target labels caused lower GAP@20 results.

Flowchart of our approach



First level models

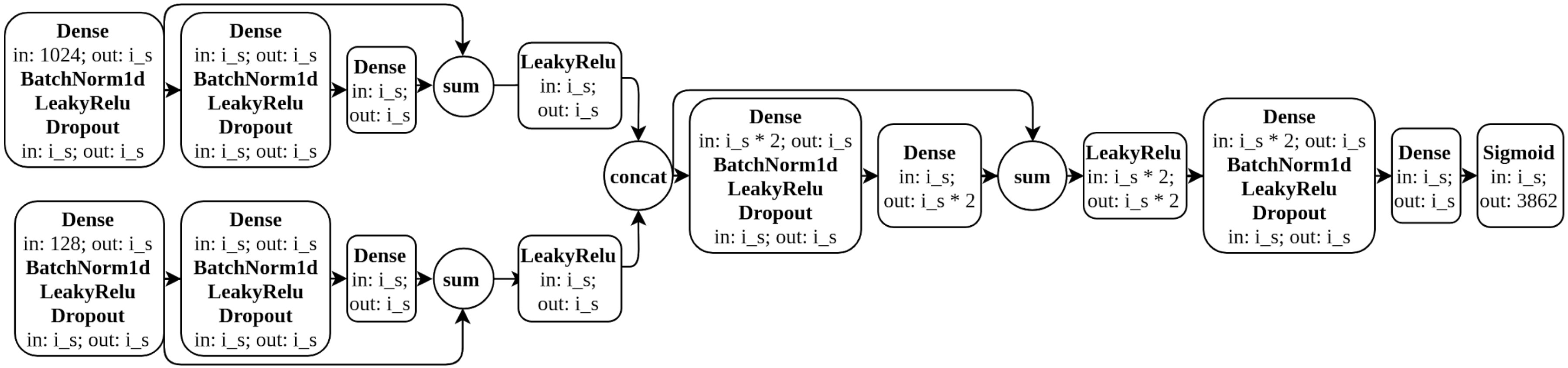
- We used only neural networks models both as for video-level and frame-level;
- Models were written in PyTorch and trained using multiple NV P40s;
- Trained for 4 days max;
- 95 video-level and 20 frame-level models were trained;
- For diversity some underperformed models were added
- (video/audio-only models, under fitted models, models trained on subsampled features, etc.)

Data aug. & Sampling

mixup; subsampling frames {at random | at regular intervals | using thresholds for cosine distances};

Video-label models

- ResNet-like architecture [n01z3]
- More than 90 different ResNet-like models were used as a first-level ensemble;
- Hyperparameters were tuned: Number of Audio & Video blocks, Inner size, Dropout.



ResNet like architecture with AV_Blocks = 1, Inner size = i_s

The best GAP@20 with ResNet-like architecture was: **0.87417** (+ soft-labels), **0.86105** (+ mixup)

youtube8m, Samsung AI Center Moscow (#4) Credits: n01z3 kaggle_yt8m https://github.com/n01zhttps://github.com/n01z3/kaggle_yt8m3/kaggle_yt8m

Frame-level models

Temporal frame-level representation of the videos was used in frame-level models

- Unidirectional and bidirectional LSTM (2x1024) followed by FC (2048);
- Learnable bag-of-words via VLADBoW model;
- Attention-based model;
- Time-distributed models (with convolution/dense layers);
- Frames replaced w/ cluster centroids (k-means, k=10000);

Best GAP@20 for single model (frame-level): **0.85325**

Second level model

We implemented several ensembling stages for the second level models:

- Second level LGBM model over top-30 categories of best first level models
- Small ensemble (6 models) trained on the out-of-fold soft-labels
- Final model trained on predictions of small ensemble in common TF Graph

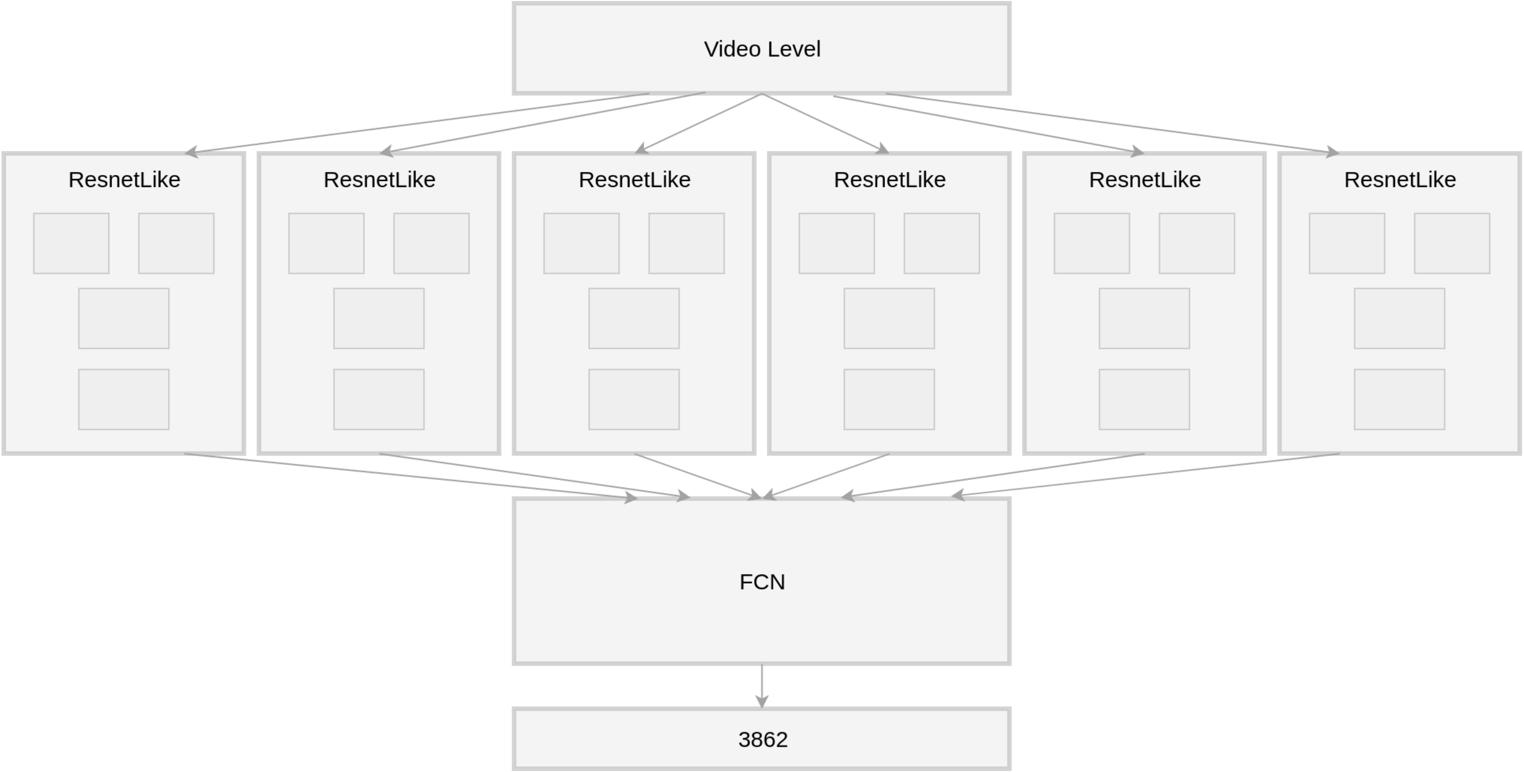
Best GAP@20 for Large Ensemble: **0.88743**
Best GAP@20 for **Final Ensemble**: **0.88729**

SAMSUNG AI Center - Moscow

LGBM dataset

	Tag 1	*Tag 2*	*Tag 3*	*Tag 4*	...	*Tag 5*
Model 1	0.99	0.97	0.96	0.965	...	0.72
Model 2	0.99	0.98	0.34	0.21	...	0.87
Model 3	0.89	0.99	0.99	0.98	...	0.71
Model ...	...	...	...	...	...	...
Model 95	0.90	0.975	0.98	0.99	...	0.7
Label	1	1	0	0		1

Final Ensemble



Details and insights

- Using frame-level models didn't show any significant improvements over video-level models (see results);
- EDA was kind of useless in the competition (at least for us);
- We assume there are still many noisy labels in the dataset;
- Lower batch size improve results, while not increasing training time;
- BCE results strongly correlate with GAP@20 evaluation results.

Results

	Model	Fr.	GAP@20	BCE	Ens.
	Final ensemble	✓	**0.88729**	—	✓
1	ResNetLike + soft labels	×	0.87417	9.2×10^{-4}	✓
2	ResNetLike + mixup	×	0.86105	9.7×10^{-4}	✓
3	ResNetLike over linear combinations	✓	0.85325	1.02×10^{-3}	✓
4	ResNetLike + soft ranking loss	×	0.85184	—	✓
5	AttentionNet	✓	0.85094	1.08×10^{-3}	✓
6	LSTM-Bi-Attention	✓	0.84645	1.04×10^{-3}	✓
7	Time Distributed Convolutions	✓	0.84144	1.0×10^{-3}	✓
8	VLAD-BOW + learnable power	✓	0.83959	1.1×10^{-3}	✓
9	Video only ResNetLike	×	0.83212	1.1×10^{-3}	✓
10	Time Distributed Dense Sorting	✓	0.83136	—	×
11	EarlyConcatLSTM	✓	0.82998	1.2×10^{-3}	✓
12	Time Distributed Dense Max Pooling	✓	0.82656	1.1×10^{-3}	✓
13	Self-attention (transformer encoder)	✓	0.8237	1.2×10^{-3}	✓
14	10000 clusters + ResNetLike	✓	0.7900	1.3×10^{-3}	✓
15	Audio only ResNetLike	×	0.50676	2.5×10^{-3}	✓
16	Bottleneck 4 neurons	×	0.41079	2.9×10^{-3}	✓

Models scores

Fr. — Frame-level models, **Ens.** — model was a part of final ensemble

Results (leaderboard)

- No shake-up;
- Starter Code gives 0.80931;
- Green / Gold / Silver / Bronze: 0.88722, 0.88324, 0.86004, 0.82930

#	△pub	Team Name	Kernel	Team Members	Score	Entries	Last
1	—	▶Next top GB model			0.88987	57	23d
2	—	[ods.ai] Evgeny Semyonov			0.88848	34	24d
3	▲1	Axon AI		+6	0.88733	51	22d
4	▲1	Samsung AI Center Moscow		+3	0.88729	66	23d
5	▼2	PhoenixLin			0.88722	41	23d
6	—	YT8M-T			0.88704	53	22d
7	—	MA			0.88664	77	25d
8	—	Licio.JL			0.88597	62	23d
9	▲1	KANU		+3	0.88527	38	23d
10	▲1	Liu			0.88324	35	24d

Second place solution overview

- Re-use prevous year competition first place solution
- Total 7 frame-level weighted mean ensemble
- Main models: NetVLAD, NetRVLAD, NetFVModel etc., also recurrent models
- Mixture of Expert as final layer
- Size reducing tricks:
  - Removing Adam weights from graph
  - Float16 weight quantization
  - Reducing number of clusters in *VLAD and *BOW models

Conclusion

- Use ensembling and distillation;
- Large ensembles can be good even if models within ensemble have weak performance;
- Soft labels can be useful when labeling is noisy;
- Mixup works.

Thank you for your attention

Vladimir Aliev @twoleggedeye
Gleb Sterkin @gleb.sterkin
Pavel Ostyakov @ostyakov
Elizaveta Logacheva @iff
Roman Suvorov @windj007
Oleg Khomenko @olegk

(#4)