Label Denoising with Large Ensembles of Heterogeneous Neural Networks

Pavel Ostyakov, Roman Suvorov, Elizaveta Logacheva,
Vladimir Aliev, Gleb Sterkin, Oleg Khomenko
{p.ostyakov, r.suvorov, e.logacheva, v.aliev, g.sterkin, o.khomenko}@samsung.com

SAMSUNG AI Center
- Moscow

(2nd place)

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.

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.

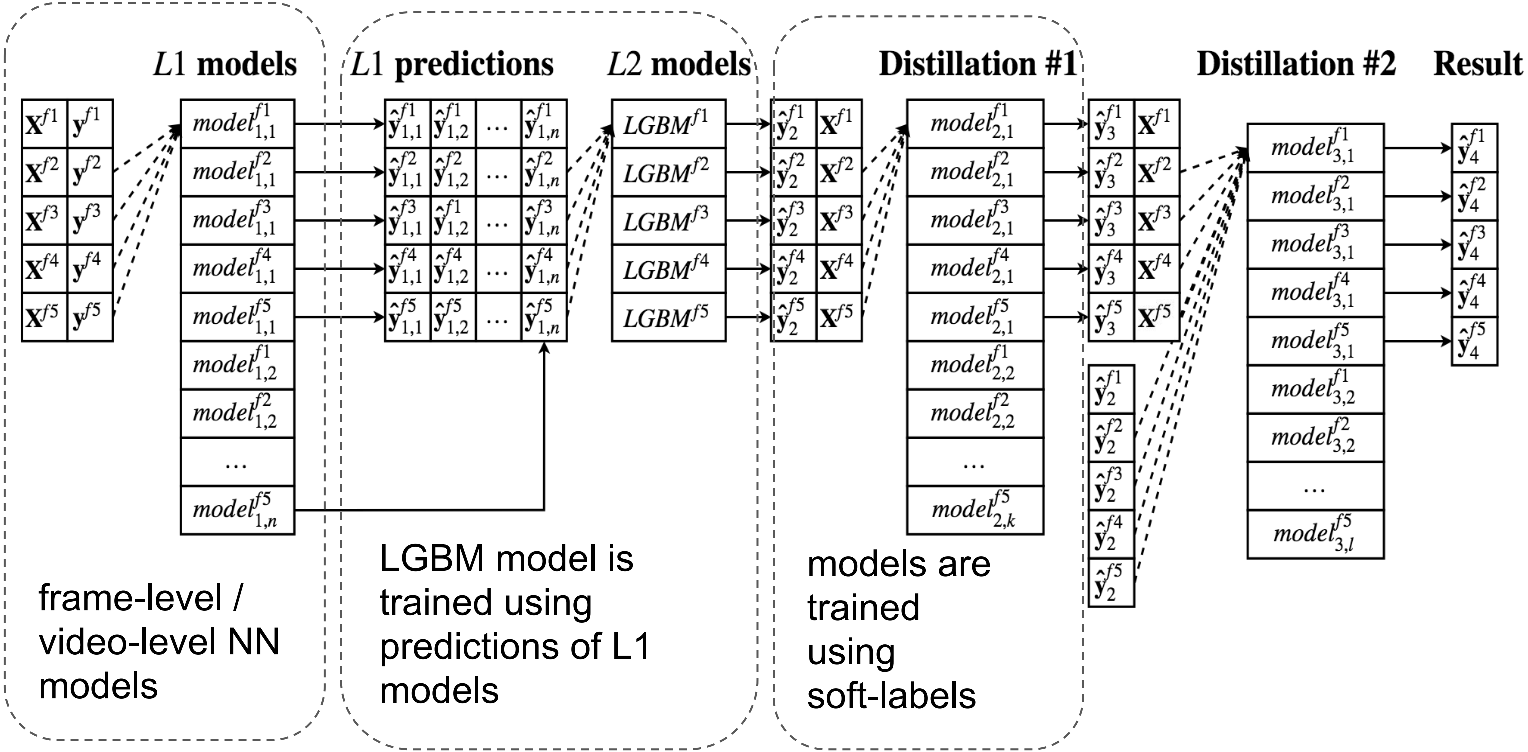
General approach

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};

MixUp

The mixup method produces “virtual” training samples as linear combinations of existing training and their targets:

$$x = \lambda x_i + (1 - \lambda) x_j$$
$$y = \lambda y_i + (1 - \lambda) y_j$$

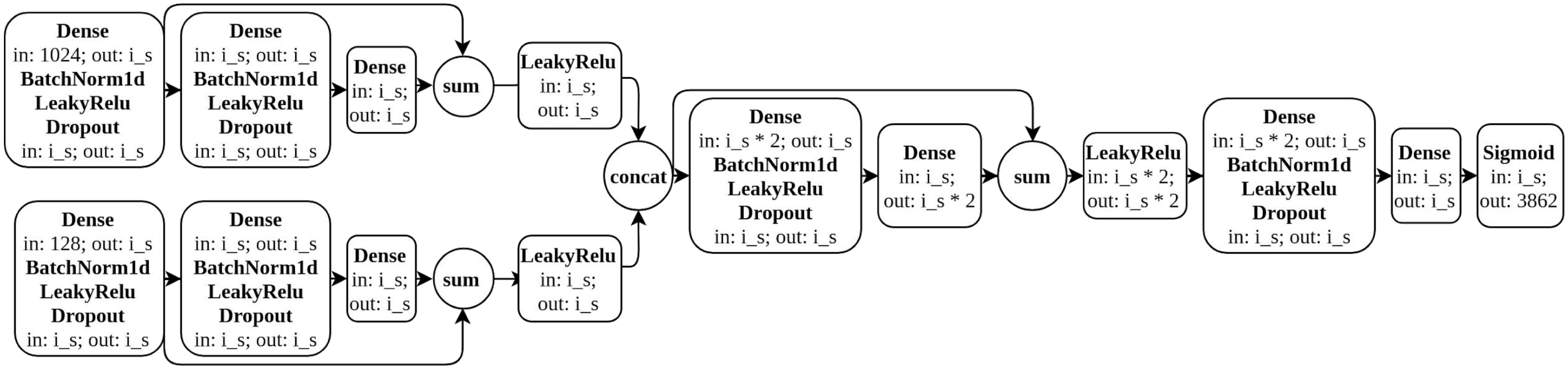
where (x_i, y_i) and (x_j, y_j) are feature-target vectors sampled from training data and

$\lambda \sim Beta(\alpha, \alpha)$, where α = 0.4 (empirically set parameter)

 Credits: Zhang, Hongyi, et al. "mixup: Beyond empirical risk minimization." *arXiv preprint arXiv:1710.09412* (2017).

Video-level 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)

Frame-level models

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

- Unidirectional and bidirectional LSTM followed by FC;
- Learnable bag-of-words via VLADBoW model;
- Attention-based model;
- Time-distributed models (with convolution/dense layers);
- Frames replaced with 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

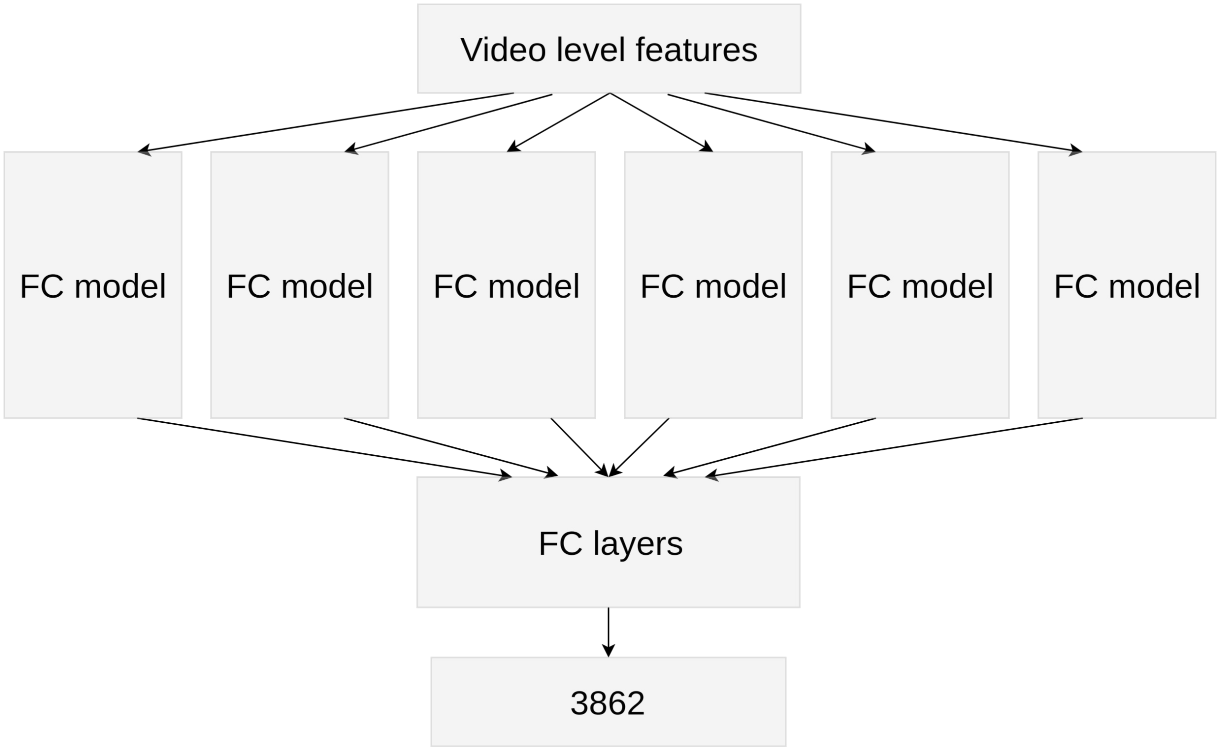
Best GAP@20 for Large Ensemble: **0.88943**

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

LGBM dataset

	Class ID	Model 1	Model 2	Model 3	...	Model 115	Label
Tag 1	34	0.99	0.97	0.975	...	0.87	1
Tag 2	3189	0.98	0.87	0.93	...	0.71	1
Tag 3	574	0.99	0.3	0.54	...	0.89	0
...	...	...	...	...	...	...	...
Tag 30	920	0.92	0.94	0.99	...	0.1	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 improves results, while not increasing training time;
- BCE results strongly correlate with GAP@20 evaluation results.

Results (validation)

	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}	✓

Validation results for models.

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: 0.88527, 0.88027, 0.86004, 0.82930

1	—	▶Next top GB model		0.88987	57	1mo
2	▲1	Samsung AI Center Moscow	+3	0.88729	66	1mo
3	▼1	PhoenixLin		0.88722	41	1mo
4	—	YT8M-T		0.88704	53	1mo
5	▲1	KANU	+3	0.88527	38	1mo
6	▲1	[ods.ai] Evgeny Semyonov		0.88506	34	1mo
7	▲1	Liu		0.88324	35	1mo
8	▲2	Sergey Zhitansky		0.88113	39	1mo
9	▲2	404 not found		0.88067	13	1mo
10	▲2	Licio.JL		0.88027	62	1mo

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

Pavel Ostyakov, Roman Suvorov, Elizaveta Logacheva,
Vladimir Aliev, Gleb Sterkin, Oleg Khomenko
{p.ostyakov, r.suvorov, e.logacheva, v.aliev, g.sterkin, o.khomenko}@samsung.com

We are hiring!

SAMSUNG AI Center
- Moscow