Efficient Population Based Data Augmentation in Speaker Recognition

Master's thesis extended summary
Author: Andres Käver (183263IAPM)
Supervisor: Tanel Alumäe, PhD

Topic explanation

Speaker recognition (identification and verification) from audio recordings is one of the many subtasks in NLP (natural language processing) field. It has wide usage in practice, from biometrical security to multi-user support in personal assistants (like Siri or Alexa) and from producing meeting transcriptions to military applications.

Over time there have been many different approaches how to achieve acceptable performance in this field. Starting initially from classical statistical methods and recently moving over to neural networks-based solutions. Most of audio based neural network research has been done in Kaldi [1] project – fairly complex system written in low-level C/C++. In last few years most of academic research in NLP field is neural network based and has been done in PyTorch (and TensorFlow) – including now also audio-based projects. Tooling and possible architectures for audio projects in neural network projects are still not well established.

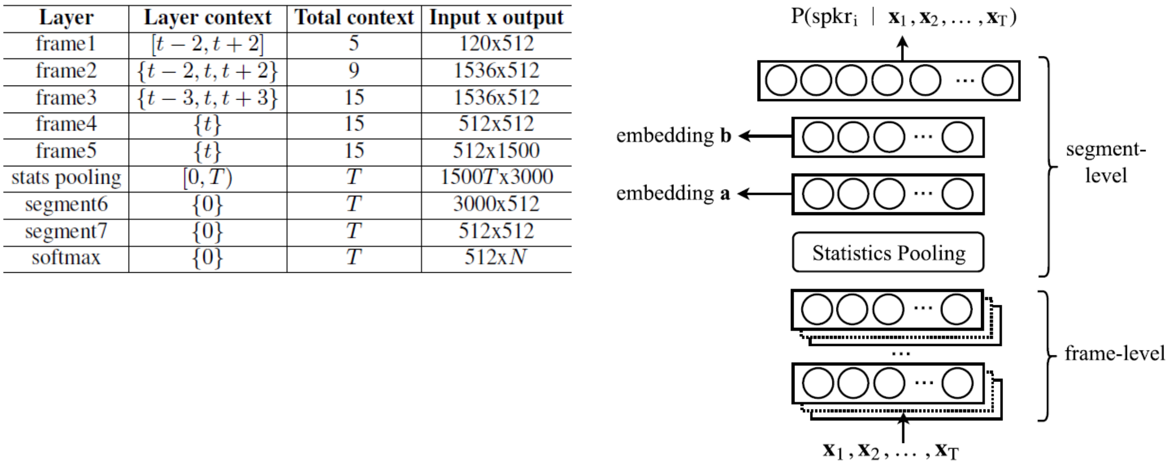
As in almost all NLP tasks, scarcity and variety of training materials is a huge problem. Typically, NLP tasks are text based, simplifying collecting training materials somewhat – at least in most widely used languages. Speaker recognition is purely audio based, thus requiring specialized datasets for training – there is only few such datasets available. One possible approach to solve this problem would be to produce such training data synthetically via modifications (augmenting) the existing training data.

Achieving state-of-the-art (sota) results in neural networks-based solutions requires high amounts of computing resources. Especially true in creating NLP universal language models – here costs of testing out various network architectures and then doing full training run has risen into millions of dollars [17]. Even somewhat simpler image recognition tasks require large amount of computing resources – often making achieving or replicating similar sota results unobtainable [16].

Lack of training materials combined with high computational resource challenges makes this research topic really challenging and interesting.

Overview of scientific publications. Problem explanation.

Initial automatic speaker identification methods were based on Gaussian mixture models (GMM) [2,3] and verification on GMM-UBM [2] (GMM and universal background model). Next evolution before deep neural networks was i-vector [4,5,6] – factor analysis was used to compute speaker- and session-dependent GMM supervector (GMM supervector is derived by concatenating the parameters of the GMM (mean vectors)).
Deep neural networks (DNN) are used to replace previous methods – to extract features similar to i-vectors and apply cosine distance or PLDA [7, 8] (probabilistic linear discriminant analysis) as decision making. Some end-to-end machine learning solutions are also proposed [9, 10].
Best results using DNN-s so far are obtained extracting features called X-vector [11, 12] – Fig. 1. It is based on DNN embeddings (taking output weights of some layer as N dimensional vector), which employs a multiple layered DNN architecture (with fully connected layers) with different temporal (time) context at each layer. Fig. 1 shows proposed X-vectors either from segment 6 (embedding a) or segment 7 (embedding b).
Various experiments have been made with input data and network architecture to obtain better results for some specific task [13, 14].



Layer	Layer context	Total context	Input x output
frame1	$[t-2, t+2]$	5	120x512
frame2	$\{t-2, t, t+2\}$	9	1536x512
frame3	$\{t-3, t, t+3\}$	15	1536x512
frame4	$\{t\}$	15	512x512
frame5	$\{t\}$	15	512x1500
stats pooling	$[0, T)$	T	1500Tx3000
segment6	$\{0\}$	T	3000x512
segment7	$\{0\}$	T	512x512
softmax	$\{0\}$	T	512xN

$P(\text{spkr}_i \mid \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)$
embedding **b**
embedding **a**
Statistics Pooling
segment-level
frame-level
$\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$

Fig. 1. X-vector DNN embedding architecture [11].

So far X-vectors are only implemented fully in Kaldi project, limiting their wider usage.

It has been experimentally shown, that augmenting/synthesizing training data improves results. Also, in what training step to use what specific augmentations also greatly influences DNN final accuracy. To figure out the best possible combination of augmentations and their schedule is very computing intensive [15].
One promising approach to explore would be Population Based Augmentation (PBA) [16].
Apply random combination of augmentations only for some (1-3) epochs – train DNN models in parallel (using reduced (ca 10x) datasets). Choose the better performing models, copy weights to lower performance models, repeat training. Stop after N steps.
Apply best found schedule to train DNN on full dataset.

Thus, the master’s thesis topic can be divided into several sub-tasks:

- Implement X-vector based system in PyTorch.
- Experiment with X-vector architecture.
- Train universal speaker embedding model – by using all the available training data and by augmenting that data. Hypothesis is, that by training with enough data it is possible to enrol further speakers with just few audio samples. This means that trained model has to be really robust – covering all common human speech audio variations. Different speakers, different languages, background noises, etc.
- Implement population-based data augmentation scheduling approach to reduce training costs to achieve sota results. Hypothesis is, that this should reduce the need for computing resources to achieve sota level results ca 1000x [16].

Methodology and work structure

First task is to implement X-vector – research and compare experiments done so far, implement baseline approach in PyTorch.

Collect suitable training data (labelled speaker utterances): Most used dataset is VoxCeleb2 – ca 6000 different celebrity speakers from YouTube videos. But there are some other possibilities also – OK Google (80 000 speakers, but all saying the same phrase), Librispeech (collected from audiobooks), NIST SRE (telephone recordings) [18, 19]. To be able to produce really universal speaker model requires as much different data samples as possible – collecting enough data is challenge in itself.

One widely used technique to save training costs is to use transfer learning. Idea is to use some pretrained universal model and retrain it for specific use-case. Using big enough training data corpus for speakers I plan to achieve similar end-result for speaker verification tasks – by training universal speaker embedding model (by extracting x-vectors from DNN hidden layers). This requires working out different model valuation then just typical straight multi-label classification – since quality of model is determined later with previously unseen speaker audio samples.

To drastically reduce training cost (and even make such training feasible with currently available limited resources – either Google Colab or authors personal machine learning computer with RTX2080 8gb card) population-based data augmentation needs to be implemented. Main goal with PBA is to implement it as reusable separate framework which can be easily used in other machine learning tasks also.

Validating

Two main measurable validation methods:

- how much less did the PBA approach used computing resources for training vs the traditional approach
- trained network is usually validated in speaker verification with following methods:
  - Cosine Distance - The cosine distance is simply computing the normalized dot product of target and test x-vectors (speaker embeddings)
  - PLDA – probabilistic linear discriminant analysis. LDA is deterministic approach to reduce the dimensionality - minimizing within-class variation and maximizing between-class variation. PLDA applies similar probabilistic approach which can be used with previously unseen examples.

Goal is to reduce training costs by factor of 1000x and achieve sota results in speaker verifications tasks – using PyTorch.

References

1. Kaldi project. https://kaldi-asr.org/doc/about.html
2. Hansen, J. HL, Hasan, T. (2015). Speaker recognition by machines and humans: A tutorial review. IEEE Signal processing magazine, 32.6, pp. 74-99.
3. Reynolds, D. A., Rose, R. C. (1995). Robust text-independent speaker identification using Gaussian mixture speaker models. IEEE transactions on speech and audio processing, 3(1), pp. 72- 83.
4. Dehak, N., Dehak, R., Kenny, P., Brümmer, N., Ouellet, P., Dumouchel, P. (2009). Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification. In: Proc. Interspeech, 2009, pp. 1559–1562.
5. Dehak, N., Kenny, P. J., Dehak, R., Dumouchel, P., Ouellet, P. (2011). Front-end factor analysis for speaker verification. IEEE Trans. Audio, Speech, Lang. Process., vol. 19, no. 4, pp. 788– 798.
6. Dehak, N., Kenny, P., Dehak, R., Glembek, O., Dumouchel, P., Burget, L., Hubeika, V., Castaldo, F. (2009). Support vector machines and joint factor analysis for speaker verification. In: Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP'09), pp. 4237–4240.
7. Tipping, M., Bishop, C. (1997). Probabilistic principal component analysis. Technical Report NCRG/97/010, Neural Computing Research Group, Aston University.
8. Ioffe, S. (2006). Probabilistic linear discriminant analysis. In: European Conference on Computer Vision. Springer, Berlin, Heidelberg, pp. 531-542.
9. Heigold, G., Moreno, I., Bengio, S., Shazeer, N. (2016). End-to-end text-dependent speaker verification. In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 5115-5119.
10. Yun, S., Cho, J., Eum, J., Chang, W., & Hwang, K. (2019). An End- to-End Text-independent Speaker Verification Framework with a Keyword Adversarial Network. In: Proc. Interspeech 2019, pp. 2923-2927.
11. Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., & Khudanpur, S. (2018). X-vectors: Robust dnn embeddings for speaker recognition. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5329- 5333
12. Fang, F., Wang, X., Yamagishi, J., Echizen, I., Todisco, M., Evans, N., Bonastre, J. (2019). Speaker Anonymization Using X-vector and Neural Waveform Models. arXiv preprint arXiv:1905.13561.
13. Jiang, Y., Song, Y., McLoughlin, I., Gao, Z., Dai L. (2019). An Effective Deep Embedding Learning Architecture for Speaker Verification. In: INTERSPEECH. 2019. pp. 4040-4044.
14. Kanagasundaram, A., Vogt, R., Dean, D., Sridharan, S., Mason, M. (2011). i-vector based speaker recognition on short utterances. In Proc. of Interspeech, 2011, pp. 2341–2344.
15. Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, Quoc V. Le (2018). AutoAugment: Learning Augmentation Policies from Data. arXiv:1805.09501.
16. Daniel Ho, Eric Liang, Ion Stoica, Pieter Abbeel, Xi Chen (2019). Population Based Augmentation: Efficient Learning of Augmentation Policy Schedules. arXiv:1905.05393.
17. Or Sharir, Barak Peleg, Yoav Shoham (2020). The Cost of Training NLP Models: A Concise Overview. arXiv: 2004.08900.

18. Joon Son Chung, Arsha Nagrani, Andrew Zisserman (2018). VoxCeleb2: Deep Speaker Recognition. arXiv:1806.05622.
19. Vassil Panayotov, Guoguo Chen, Daniel Povey and Sanjeev Khudanpur, ICASSP (2015). LibriSpeech: an ASR corpus based on public domain audio books. http://www.danielpovey.com/files/2015_icassp_librispeech.pdf