Speaker Identification Using Deep Neural Networks

Fan Hu (fanh), Linwu Zhong (linwuz), Siyuan Wang (siyuanw), Zinian Zhao (zinianz)

Carnegie Mellon University, 11-785 Project

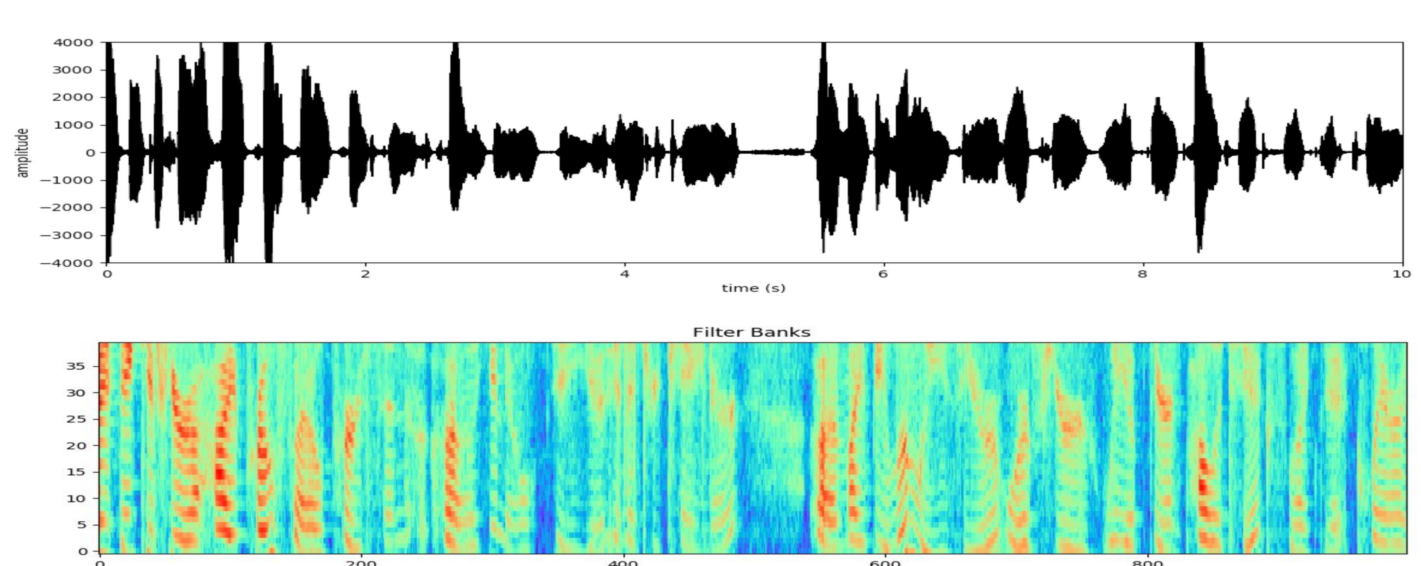
Background

In this project, we would like to work on NIST dataset and build a deep neural network (DNN) model to identify if two speech recordings belong to the same speaker. The project scope includes finding optimal feature representations from the acoustic data, exploring various advanced neural networks for the model, and achieving state-of-the-art accuracy.

Feature Extraction

In this project, the feature extraction consists of 8 main steps.

- **Voice Activity Detection**: Use WebRTC Voice Activity Detector (VAD) to remove unvoiced signal.
- **Pre-emphasis**: Pre-emphasis to balance freq spectrum as high freq components often have smaller magnitude.
- **Framing**: Divide signal into short time frames with an overlap to keep the frequency contours.
- **Windowing**: Apply Hamming window function to each frame to smoothen the signal.
- **FFT and Power Spectrum**: Apply FFT to convert signal from time domain to freq domain and calculate power spectrum.
- **Mel Spectrogram & Filter Banks**: Apply filters on a Mel-scale to extract frequency bands.
- **Mean Normalization**: Perform mean normalization on the mel-scaled filter banks to improve the Signal-to-Noise (SNR).
- **Feature Length Standardization**: Fix the time dimension in the features as 30000 by appending duplicates.



Datasets

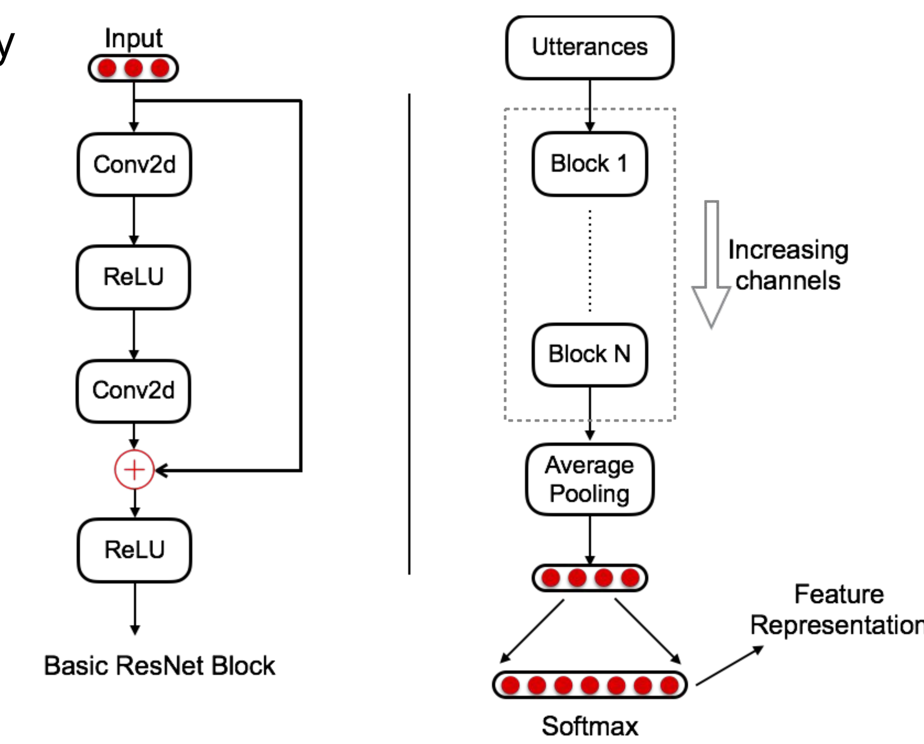
The data used in this project are NIST Speaker Recognition Evaluation (SRE) datasets from 4 different years: 2004 (12.25%), 2005 (7.47%), 2006 (50.10%), 2008 (30.18%).

There are a total of 33687 speech files. 60.23% of the files belong to female speakers, and the other 39.97% belong to male speakers.

Our Approach

Network Architecture

- Network is formed by concatenating multiple Resnet basic blocks.
- Each block contains two Conv 2d layers.
- Two different sized models to compared performance.



layer name	kernel/structure	stride
conv16	5×5	2×2
res16	3×3	1×1
conv32	5×5	2×2
res32	3×3	1×1
conv64	5×5	2×2
res64	3×3	1×1
conv128	5×5	2×2
res128	3×3	1×1
conv256	5×5	2×2
res256	3×3	1×1
conv512	5×5	2×2
res512	3×3	1×1
mean pooling	along time dim	-
linear	512×classes	-

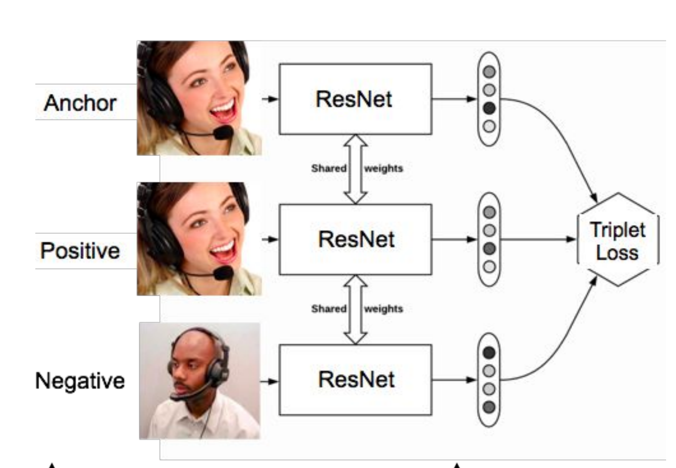
layer name	kernel/structure	stride
conv32	5×5	4×4
res32	3×3	1×1
conv128	5×5	4×4
res128	3×3	1×1
conv512	5×5	4×4
res512	3×3	1×1
mean pooling	along time dim	-
linear	512×classes	-

Table 1: Model A (large) and Model B (small)

Softmax Pre-training

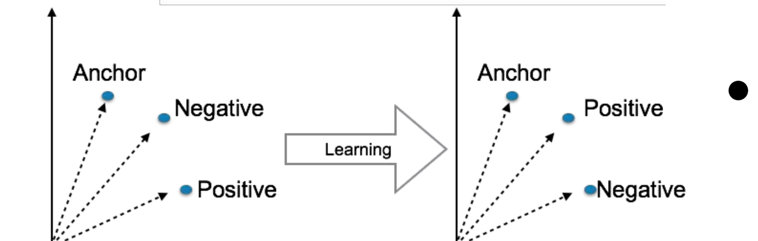
- A multi-way classification (Number of Speakers in training data)
- Used to initialize weights of the networks:
  - Cross entropy loss is stabler than triplet loss

Triplet Loss Training



$$\mathcal{L} = max(d(a,p) - d(a,n) + margin, 0)$$

- Anchor, positive, negative pairs
- Cosine distance between two representations
- Select only “hard” triplets that *d(a, p) - d(a, n) > 0* for training



Experiments & Results

Pre-training: N-way Classification

	Model A (large)	Model B (small)
Dev Accuracy	86.0%	84.4%

Triplet Loss Training

- Evaluation Metric: Equal Error Rate (EER)

Experiment 1:

- only sample training triplets from wrong classifications

	Epoch 1	Epoch 7
Train EER	30.6%	14.0%
Test EER	59.3%	40.0%

Experiment 2:

- sample training triplets from all training data (but more weight for wrong classifications)

Train EER	Test EER
11%	36%

Experiment 3:

- Add BatchNorm after average pooling

	Epoch 1	Epoch 6
Test EER	59.3%	20.4%

Conclusion

Discussion

- Speaker classification and identification are two different tasks, high classification accuracy $\nRightarrow$ good identification performance
- BatchNorm reduces covariance shift and normalizes the feature vector to a comparable range => cosine similarity more consistent
- Triplet network is hard to train, which requires careful sampling of training triplets and little tricks in neural networks

Future Work

- Experiment with RNN or other CNN architectures
- Experiment with more (maybe adaptive) sampling methods for training triplets
- Experiment with different margin values

References

[1] Najim Dehak, Patrick J. Kenny, Reda Dehak, Pierre Dumouchel, and Pierre Ouellet. Front-end factor analysis for speaker verification. IEEE Transactions on Audio, Speech, and Language Processing, 19:788–798, 2011.
[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770– 778, 2016.
[3] Patrick Kenny. Bayesian speaker verification with heavy-tailed priors. Odyssey, 2010.
[4] Hung-Shin Lee, Yu-Ding Lu, Chin-Cheng Hsu, Yu Tsao, Hsin-Min Wang, and Shyh-Kang Ieng. Discriminative autoencoders for speaker verification. Acoustics, Speech and Signal Processing (ICASSP), pp. 5375–5379, 2017.
[5] Yun Lei, Nicolas Scheffer, Luciana Ferrer, and Mitchell McLaren. A novel scheme for speaker recognition using a phonetically-aware deep neural network. Acoustics, Speech and Signal Processing (ICASSP), pp. 1695–1699, 2014.
[6] Chao Li, Xiaokong Ma, Bing Jiang, Xiangang Li, Xuewei Zhang, Xiao Liu, Ying Cao, Ajay Kannan, and Zhenyao Zhu. Deep speaker: an end-to-end neural speaker embedding system. 05 2017.
[7] Timur Pekhovsky, Sergey Novoselov, Aleksei Sholohov, and Oleg Kudashev. On autoencoders in the i-vector space for speaker recognition. Proc. Odyssey, pp. 217–224, 2016.
[8] Douglas A. Reynolds and Richard C. Rose. Robust text-independent speaker identification using gaussian mixture speaker models. IEEE Transactions on Speech and Audio Processing, 3:72–83, 1995.
[9] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. pp. 815–823, 06 2015.