GENERALIZED END-TO-END LOSS FOR SPEAKER VERIFICATION

SUPERVECTOR

JFA

I-VECTOR

D-VECTOR

TE2E

GE2E

GE2E : ABSTRACT

이것을 선택한 이유는?

- 화자 인식 vs 얼굴 인식
- 비슷한 점이 있을까 해서 찾아봤음

GE2E : ABSTRACT

화자 인식 분야의 새로운 loss 설계 : generalized end-to-end (GE2E) loss

- decreases speaker verification EER by more than 10%
- reducing the training time by 60%

"OK Google" and "Hey Google" 에서 화자 구분 할때 사용

GE2E : INTRODUCTION

Speaker Verification(SV)

- text-dependent speaker verification (TD-SV)
- text-independent speaker verification (TI-SV).

* 교재 참고

Text-dependent System

- Highly constrained (fixed or prompted) text samples
- Used for applications with strong control over user input
- Knowledge of spoken text characteristic can improve system performance

Text-independent System

- Unconstrained (user selected or conversational speech) text samples
- Used for applications with less control over user input
- More flexible system but more difficult problem
- Speech recognition can additionally provide knowledge of spoken text

GE2E : INTRODUCTION

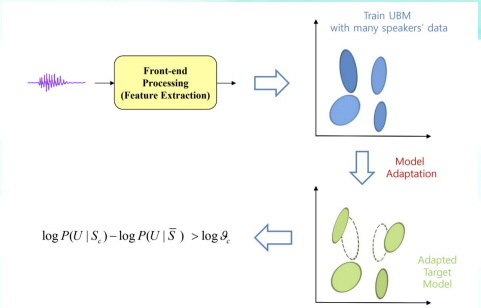
Background : super vector

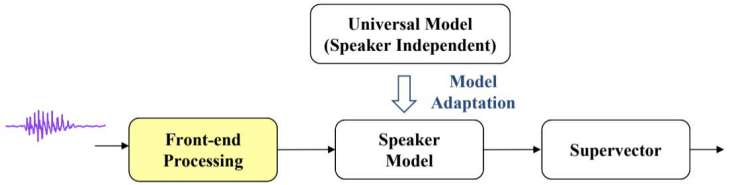
- 시작은 GMM-UBM Speaker Recongnition에서 시작 * gmm : Gausian Mixture Model

  1) UBM (Universial background model) 훈련 후

  2) 각 Speaker별로 Adaptation(ML를 쓰던지, MAP를 쓰던지)

  3) 그 다음에 SuperVector를 뽑음





GE2E : INTRODUCTION

Background : super-vector

- SuperVector ? Speaker dependent GMM mean components

❖ Sufficient statistics to compute the likelihood of GMM

$$\mathbf{g}_i = \sum_{n=1}^{N} \Pr(i \mid \mathbf{x}_n) \qquad E_i(\mathbf{x}) = \sum_{n=1}^{N} \Pr(i \mid \mathbf{x}_n)\mathbf{x}_n$$

$$\mathbf{m}_i = \frac{E_i(\mathbf{x})}{\mathbf{g}_i} \qquad \mathbf{S}_i = \frac{E_i(\mathbf{x}\mathbf{x}^T)}{\mathbf{g}_i}$$

i: mixture index
$\mathbf{x}_n$: target training data
N: # of training data
M: # of mixtures

❖ Log likelihood of mean adaptation only case

$$E\left[\log p(X, I \mid \lambda)\right] = -\frac{1}{2}\sum_{i=1}^{M} (\mathbf{m}_i - \boldsymbol{\mu}_i)^T \left(\frac{\boldsymbol{\Sigma}_i^{ubm}}{\mathbf{g}_i}\right)^{-1} (\mathbf{m}_i - \boldsymbol{\mu}_i) + C_i$$

$$\left[\begin{bmatrix}\mathbf{m}_1\\ \mathbf{m}_2\\ \vdots\\ \mathbf{m}_M\end{bmatrix} - \begin{bmatrix}\boldsymbol{\mu}_1\\ \boldsymbol{\mu}_2\\ \vdots\\ \boldsymbol{\mu}_M\end{bmatrix}\right]^T \begin{bmatrix}\boldsymbol{\Sigma}_1^{-1} & & & \mathbf{O}\\ & \boldsymbol{\Sigma}_2^{-1} & & \\ & & \ddots & \\ \mathbf{O} & & & \boldsymbol{\Sigma}_M^{-1}\end{bmatrix} \left[\begin{bmatrix}\mathbf{m}_1\\ \mathbf{m}_2\\ \vdots\\ \mathbf{m}_M\end{bmatrix} - \begin{bmatrix}\boldsymbol{\mu}_1\\ \boldsymbol{\mu}_2\\ \vdots\\ \boldsymbol{\mu}_M\end{bmatrix}\right]$$

* supervectors: stack vectors and matrices

평균과 ubm에서 얻었던 차이를 모아 논것을 SuperVector라고 함

*** 자세한것은 음성인식 Study의 자료 참고**

- 근데 SuperVector는 스피커의 특성외에, 주변 다른 채널의 정보도 같이 있음
- 이걸 분리 해볼까?

GE2E : INTRODUCTION

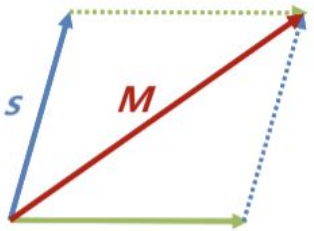
Background : Joint Factor Analysis(JFA)

- JFA로 분리
  SuperVector를 공간에 잘 놓은다음 한축은 화자의 정보 한축은 환경에 대한 정보로 분리하여 화자의 정보만 사용함

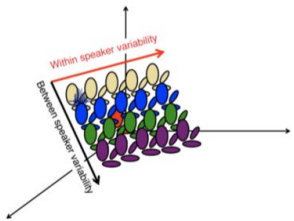
❖ Supervector $M = s + c = m + Vy + Dz + Ux$

- Speaker-dependent + Channel-dependent

- Speaker dependent: $s = m + Vy + Dz$



P. Kenny, “Joint factor analysis of speaker and session variability: Theory and algorithms,” CRIM, Montreal, CRIM-06/08-13, 2005.



- 결국 벡터를 화자(S) + 채널(C)로 구분하겠다는건데 잘 분리가 안됨.
  그래서 i Vector가 나옴

GE2E : INTRODUCTION

Background : I-Vector(Speaker Identity Vector)

- I-Vector intermediate Vector

  - Low-dim. speaker- and channel-dependent space using a factor analysis

$$M = m + Tw$$

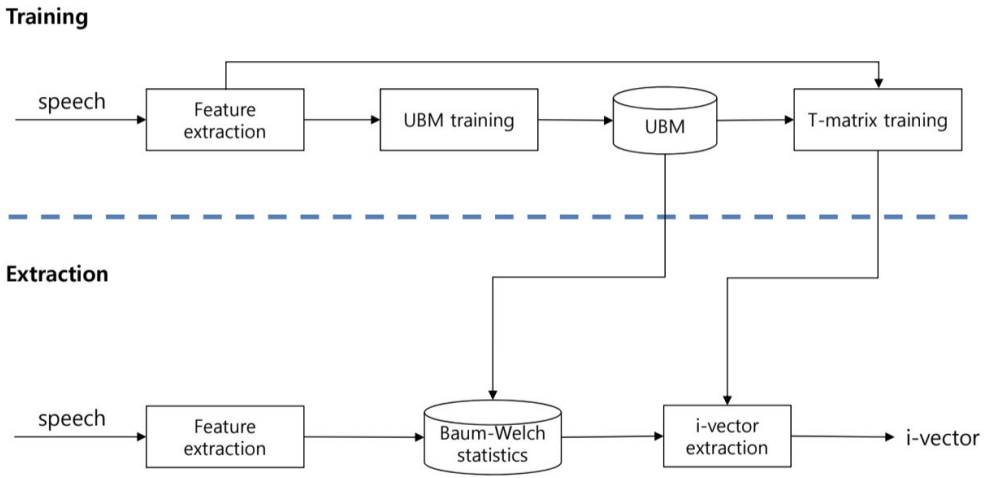
i-vector

T: rectangular matrix of low rank (total variability matrix)
w: random vector having a standard normal distribution (i-vector)

- m(유니버설모델) + T(스피커 정보를 다 포함안 공간) w(스피커의 위치) 이런 느낌임.
- 개념적으로 T는 스피커의 정보를 다 포함한 공간(채널과 스피커 정보 포함)에서 특정 스피커의 위치를 i-vector라고 함(수식은 교재 참고)
- I vector는 JFA와 비슷한 방법이지만 JFA방식은 화자 단위로 음성 데이터를 처리하는 반면, i-vector는 발성 단위로 음성 데이터를 처리 한다.

GE2E : INTRODUCTION

Background : I-Vector(Speaker Identity Vector)



https://brunch.co.kr/@kakao-it/135

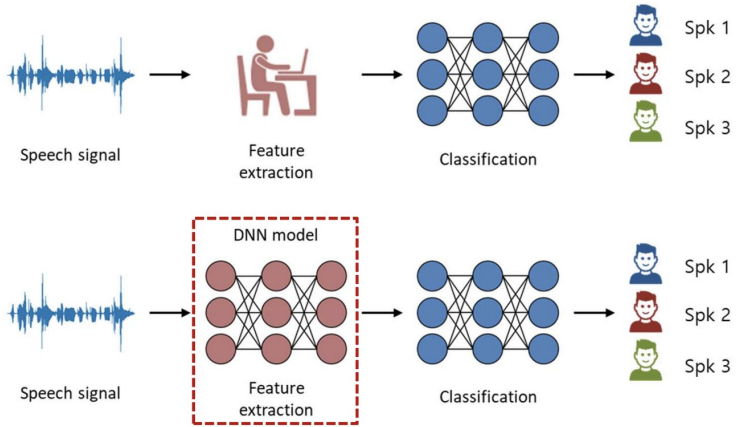
N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, “Front-end factor analysis for speaker verification,” IEEE Transactions on Audio, Speech, and Language Processing, vol. 19, no. 4, pp. 788–798, 2011.

GE2E : INTRODUCTION

Background : d-vector

- **Conventional methods :**
  I-vector(Deep learning 나오기전 SV에서 SOTA)
  GMM Super vector

- **Deep learning based methods:**
  D-vector, X-vector



Ehsan Variani, Xin Lei, Erik McDermott, Ignacio Lopez Moreno, and Javier Gonzalez-Dominguez, “Deep neural networks for small footprint text-dependent speaker verification,” in Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on. IEEE, 2014, pp. 4052– 4056.

GE2E : INTRODUCTION

Background : d-vector

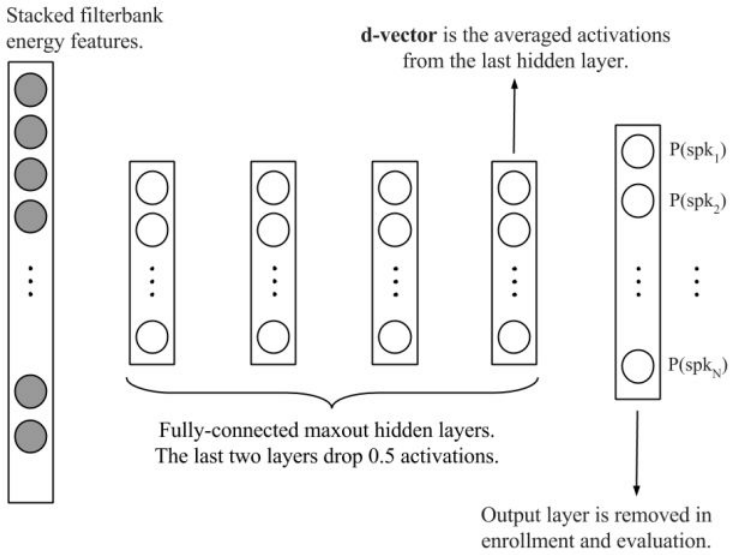


Fig. 1. The background DNN model for speaker verification.

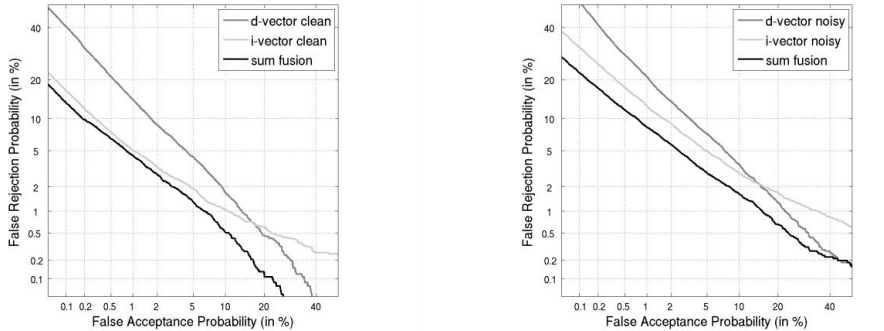


Fig. 3. DET curve for the sum fusion of the *i*-vector and *d*-vector systems in clean (left) and noisy (right) conditions.

The EER of the combined system is 14% and 25% better than our classical i-vector system in clean and noisy conditions respectively.

GE2E : INTRODUCTION

Tuple-Based End-to-End Loss : D-vector for speaker verification

- **DNN as a feature extractor**
  last hidden layer using standard feedforward propagation in the trained DNN, and then accumulate those activations to form a new compact representation of that speaker, the **d-vector**

- **Enrollment and evaluation**
  the **cosine distance** between the test d-vector and the claimed speaker's d-vector.

CNN, LSTM으로 발전, 그리고 End 2 End로

GE2E : INTRODUCTION

Tuple-Based End-to-End Loss : TE2E

For each input tuple, we compute the L2 normalized response of the LSTM: {ej~,(ek1, . . . , ekM)}.

$$\mathbf{c}_k = \mathbb{E}_m[\mathbf{e}_{km}] = \frac{1}{M}\sum_{m=1}^{M}\mathbf{e}_{km}. \quad (1)$$

The similarity is defined using the cosine similarity function:

$$s = w \cdot \cos(\mathbf{e}_{j\sim}, \mathbf{c}_k) + b, \quad (2)$$

with learnable w and b. The TE2E loss is finally defined as:

$$L_{\mathrm{T}}(\mathbf{e}_{j\sim}, \mathbf{c}_k) = \delta(j,k)\sigma(s) + \Big(1-\delta(j,k)\Big)\Big(1-\sigma(s)\Big). \quad (3)$$

Cosine Similarity : 같으면 1, 90도면 0, 반대면 -1

델타는 1과 0, 같으면 1, 다르면 0

Georg Heigold, Ignacio Moreno, Samy Bengio, and Noam Shazeer, “End-to-end text-dependent speaker verification,” in Acoustics, Speech and Signal Processing (ICASSP), 2016

GE2E : INTRODUCTION

Tuple-Based End-to-End Loss : End-to-end text-dependent speaker verification

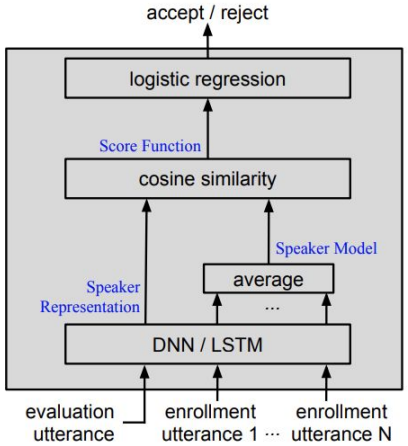


Figure 2: *End-to-end architecture: the input is an "evaluation" utterance and up to N "enrollment" utterances, which the network maps to a single output node (accept/reject). The "enrollment" utterances are used to estimate the speaker model.*

1) Tupple 만들고, (Evaluation, Enrollment)

2) Enrollment는 average

3) Cosine similarity

4) Logistic regression

Finally, we compute the cosine similarity between the speaker representation and the speaker model, $S(X, spk)$, and feed it to a logistic regression including a linear layer with a bias. The architecture is optimized using the end-to-end loss

$$l_{e2e} = -\log p(target) \quad (1)$$

with the binary variable $target \in \{\text{accept}, \text{reject}\}$, $p(\text{accept}) = (1+\exp(-wS(X, spk)-b))^{-1}$, and $p(\text{reject}) = 1-p(\text{accept})$. The value $-b/w$ corresponds with the verification threshold.

결국 Enrollment와 verification이 한번에 end to end

GE2E : GENERALIZED END-TO-END MODEL

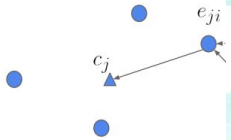
Training Methond

1) 데이터 준비 : We fetch N × M utterances to build a batch. (N different Speaker, M은 utterances)
2) 임베딩 얻기 : x_ji (input data) -> LSTM -> e_ji, f(x,w)가 lstm

$$\mathbf{e}_{ji} = \frac{f(\mathbf{x}_{ji};\mathbf{w})}{||f(\mathbf{x}_{ji};\mathbf{w})||_2}.$$

3) 임베딩 비교 : S(cosine similarity), **같으면 1, 다르면 작아짐**

$$\mathbf{S}_{ji,k} = w \cdot \cos(\mathbf{e}_{ji}, \mathbf{c}_k) + b,$$



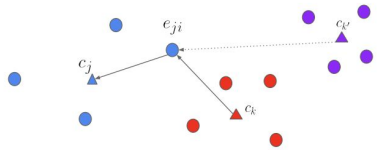
여기까지가 TE2E랑 같음

SoftMax and Contrast

1) Softmax

$$L(\mathbf{e}_{ji}) = -\mathbf{S}_{ji,j} + \log \sum_{k=1}^{N} \exp(\mathbf{S}_{ji,k}). \qquad (6)$$

Positive : 다른 스피커의 중심점에서 멀리 pull it away from all other centroids.
Negative : 각각의 임베딩은 중심과 가깝게 we push each embedding vector close to its centroid



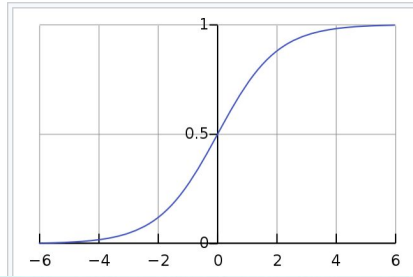
앞의 식은 중심점이랑 임베딩이 가까우면 작아지고, 뒤의 식은 다른 스피커의 중심점과 멀수록 값이 작아지고, 클스록 커짐

GE2E : GENERALIZED END-TO-END MODEL

SoftMax and Contrast

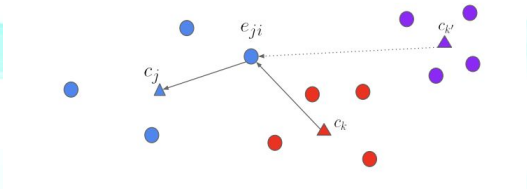
2) Contrast : 약간 Hard sample과 같은 느낌

$$L(\mathbf{e}_{ji}) = 1 - \sigma(\mathbf{S}_{ji,j}) + \max_{\substack{1 \le k \le N \\ k \neq j}} \sigma(\mathbf{S}_{ji,k}), \qquad (7)$$



Positive : 가장 가까운 화자와 멀게 pull away from closest false speaker centroid
Negative : embedding to be placed near its centroid

결국 Hard sample을 식으로 표현한것 내가 아닌 가장 가까운 클러스터의 중심점과 비교해서 멀어지게 한다.



GE2E : GENERALIZED END-TO-END MODEL

SoftMax and Contrast loss : GE2E Loss

$$\mathbf{c}_j^{(-i)} = \frac{1}{M-1}\sum_{\substack{m=1 \\ m\neq i}}^{M}\mathbf{e}_{jm}, \tag{8}$$

$$\mathbf{S}_{ji,k} = \begin{cases} w\cdot\cos(\mathbf{e}_{ji},\mathbf{c}_j^{(-i)})+b & \text{if} \quad k=j; \\ w\cdot\cos(\mathbf{e}_{ji},\mathbf{c}_k)+b & \text{otherwise.} \end{cases} \tag{9}$$

Combining Equations 4, 6, 7 and 9, the final GE2E loss L_G is the sum of all losses over the similarity matrix ($1 \leq j \leq N$, and $1 \leq i \leq M$):

$$L_G(\mathbf{x};\mathbf{w}) = L_G(\mathbf{S}) = \sum_{j,i} L(\mathbf{e}_{ji}). \tag{10}$$

앞의 개념을 잘 정리해서 10번이 나옴.
10번을 풀어 쓰면,
Loss_g = softmax loss + comparison(contrast) loss

GE2E : COMPARE TE2E AND GE2E

Comparison between TE2E and GE2E

$$2 \times \max\left(\binom{M}{P}, (N-1)\binom{M}{P}\right) \geq 2(N-1). \qquad (11)$$

M : utterance, N : Speaker, P ; enrollment

위 식에 따라 GE2E가 훨씬 빨리 학습 됨

GE2E：Multi Reader

Training with MultiReader

- **데이터가 불균형 할때 고안한 방법**
  예를 들어 Ok google은 150M, Hey google은 1.2M 밖에 수집이 안되어 있음

- **두데이터를 효과적으로 섞어서 학습 하고, Loss로 약간 다르게 변경**

$$L(D_1, D_2; \mathbf{w}) = \mathbb{E}_{x \in D_1}[L(\mathbf{x}; \mathbf{w})] + \alpha \mathbb{E}_{x \in D_2}[L(\mathbf{x}; \mathbf{w})]. \quad (12)$$

- **D1은 데이터가 적음 그래서 오버피팅이 될수 있음, D2는 데이터가 많음 D2데이터를 regularization term으로 활용하여 로스 수정**

- **D2는 알파를 웨이트로 해서 데이터셋이 여러개일때 가중치를 학습하도록 함**
-
- **Multiple keyword ,Multiple gender/age/race , Dialect and accented speech**

GE2E : GENERALIZED END-TO-END MODEL

EXPERIMENT

Table 1. MultiReader vs. directly mixing multiple data sources.

Test data (Enroll → Verify)	Mixed data EER (%)	MultiReader EER (%)
OK Google → OK Google	1.16	0.82
OK Google → Hey Google	4.47	2.99
Hey Google → OK Google	3.30	2.30
Hey Google → Hey Google	1.69	1.15

Table 2. Text-dependent speaker verification EER.

Model Architecture	Embed Size	Loss	Multi Reader	Average EER (%)
(512,) [13]	128	TE2E	No	3.30
			Yes	2.78
$(128, 64) \times 3$	64	TE2E	No	3.55
			Yes	2.67
$(128, 64) \times 3$	64	GE2E	No	3.10
			Yes	2.38

It is also worth noting that the GE2E model took about 60% less training time than TE2E.

GE2E : GENERALIZED END-TO-END MODEL

CONCLUSIONS

In this paper, we proposed the generalized end-to-end (GE2E) loss function to train speaker verification models more efficiently. Both theoretical and experimental results verified the advantage of this novel loss function.

THE END