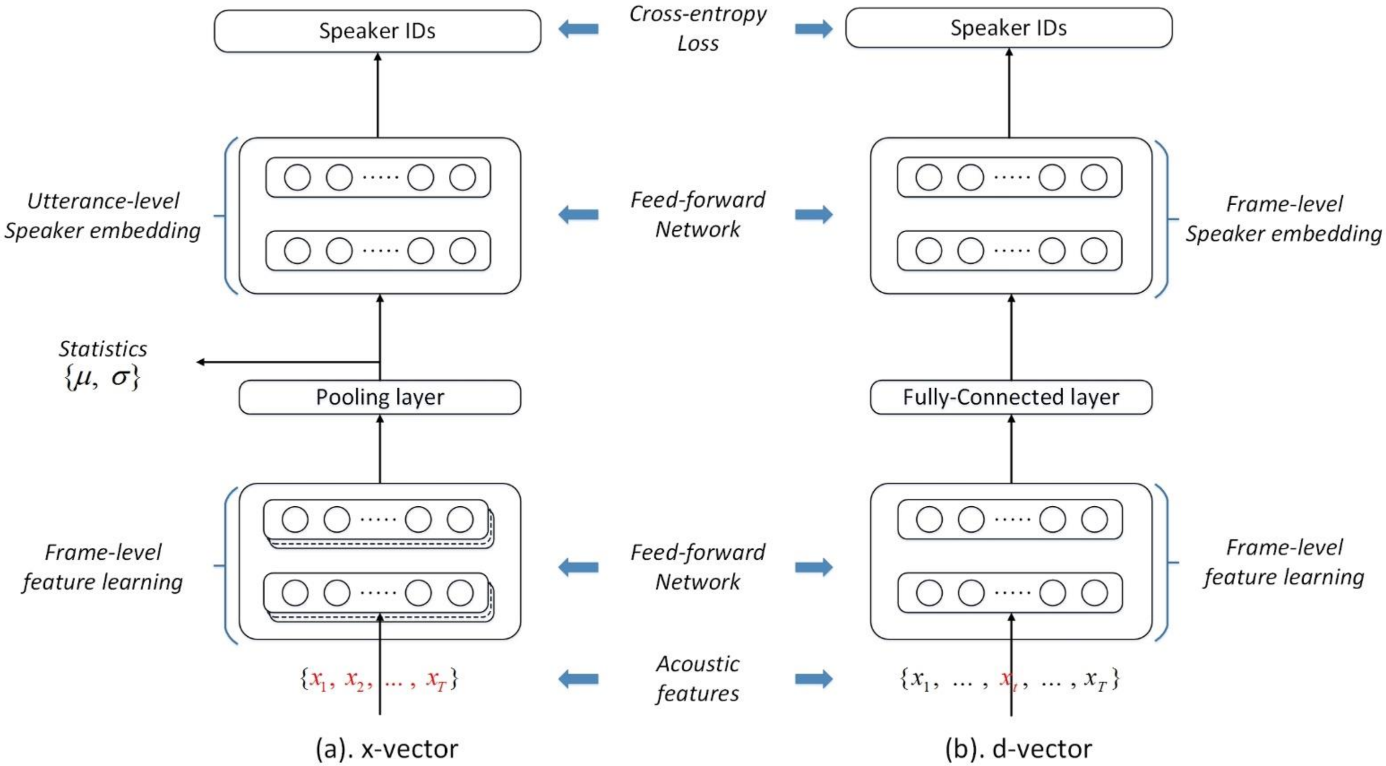
# Real Additive Margin Softmax for Speaker Verification

Lantian Li

2021.10.25

# Neural-based speaker embedding




**Classifier**

**Representation**

**Feature Extractor**

(a). D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, “X-vectors: Robust DNN embeddings for speaker recognition,” in ICASSP. IEEE, 2018.

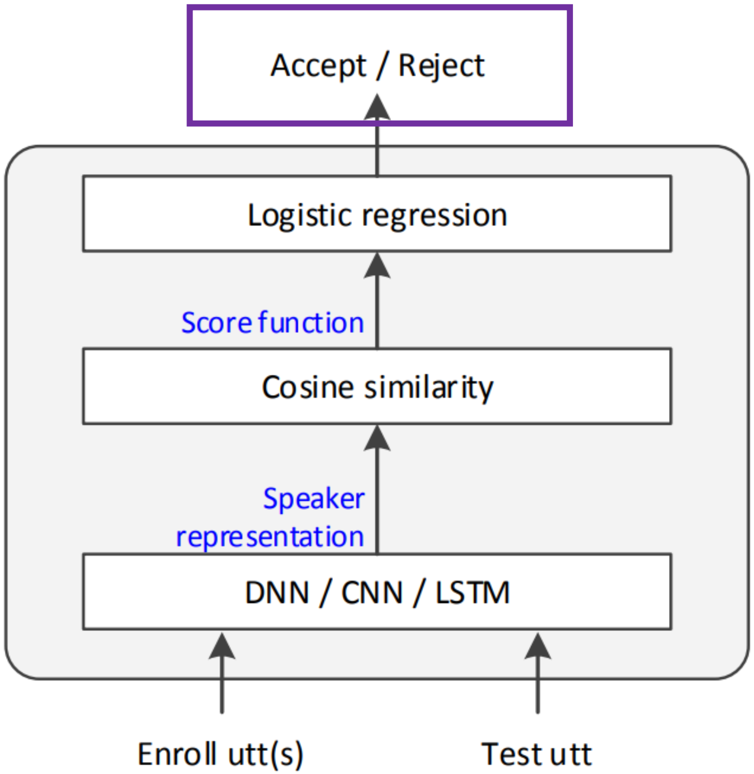
(b). E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, “Deep neural networks for small footprint text-dependent speaker verification,” in ICASSP. IEEE, 2014.

# Properties

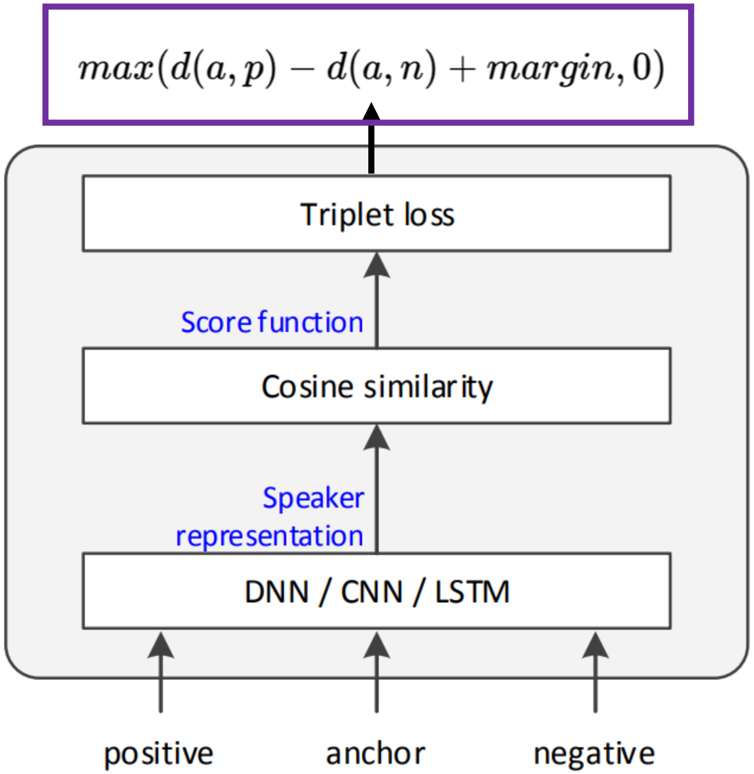
- A canonical classification framework
  - *Softmax + Cross-entropy*

- Pros
  - Optimal for discriminating speakers in the *training* set.
  - Optimal for the *close-set* ASV task.
- Cons
  - Not guaranteed on *unseen* speakers.
  - Not optimal for the *open-set* ASV task.

# Metric learning for open-set ASV




(a) Logistic regression in cosine similarity




(b) Triplet loss in cosine similarity


(a). G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in ICASSP. IEEE, 2016, pp. 5115–5119.

(b). C. Zhang and K. Koishida, "End-to-end text-independent speaker verification with triplet loss on short utterances," in INTERSPEECH, Stockholm, Sweden, 2017.

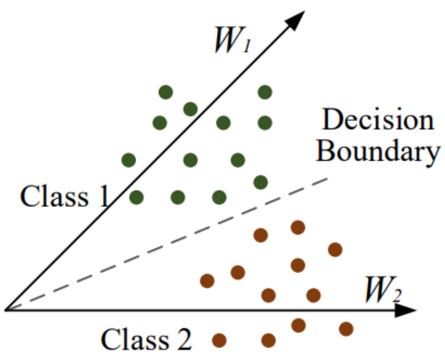
# Properties

- A canonical metric learning framework
  - Intra-speaker distance < Inter-speaker distance

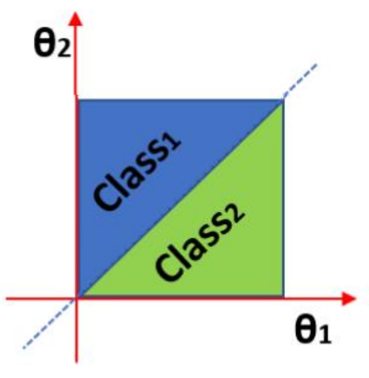
- Pros
  - *Local* difference instead of *global* discrimination
  - Optimal for the *open-set* ASV task.
- Cons
  - *Combinatorial explosion* for pairs/triplets.
  - *Difficult* for model training, e.g., local optimum or non-convergence.

# Modified softmax training

- Motivation
  - *Softmax*: simple form and easy training.
  - *Softmax* does not explicitly encourage inter-speaker separability and intra-speaker compactness.




**Original Softmax**




**Softmax**

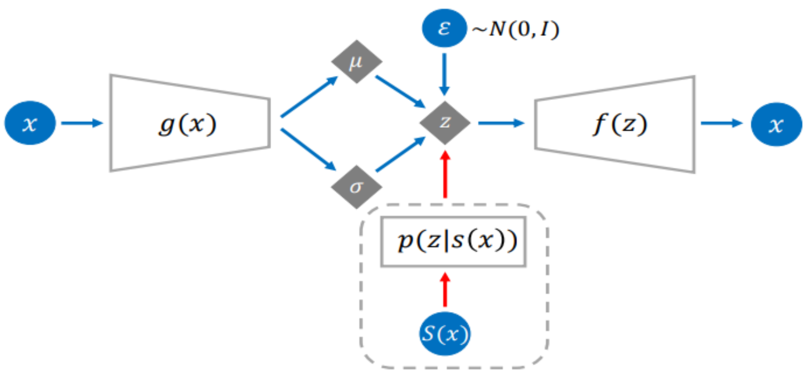
  - Produced embeddings are not generalizable to unseen speakers.

# Distribution regularization

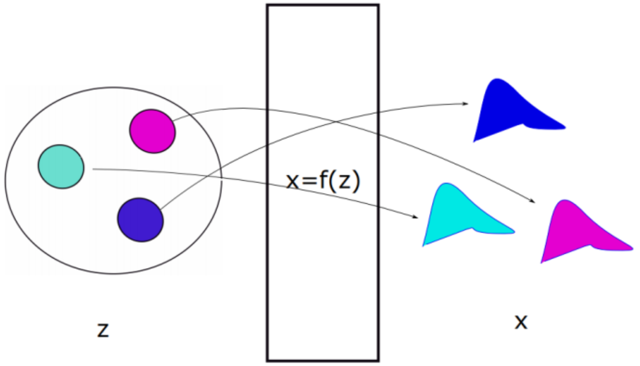
- Center loss

$$\mathcal{L}_C = \frac{1}{2}\sum_{i=1}^{m} \|\boldsymbol{x}_i - \boldsymbol{c}_{y_i}\|_2^2$$

- VAE




- DNF/NDA



# Margin-based softmax

- Softmax

$$L_S = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{\boldsymbol{w}_{y_i}^T \boldsymbol{x}_i}}{\sum_{j=1}^{C} e^{\boldsymbol{w}_j^T \boldsymbol{x}_i}}$$

- Modified Softmax

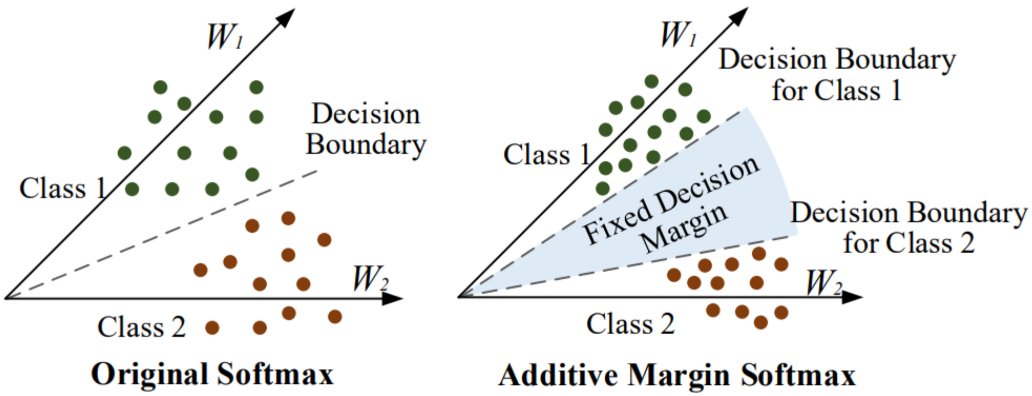
$$L_{MS} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{s \boxed{\cos\theta_{y_i}}}}{\sum_{j=1}^{C} e^{s\cos\theta_j}} \qquad ||w_j|| = ||x_i|| = 1$$

- Margin-based Softmax

$$L_{LMS} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{s\cdot \boxed{\psi(\theta_{y_i})}}}{e^{s\cdot\psi(\theta_{y_i})} + \sum_{j=1, j\neq i}^{C} e^{s\cdot\cos\theta_j}}$$

# Margin-based softmax

- Involving a fixed margin region in the target logit.




$$\psi(\theta_{y_i}) = \cos(m_1\theta_{y_i} + m_2) - m_3$$

- m1: angular softmax (*A-Softmax*)
- m2: additive angular margin softmax (*AAM-Softmax*)
- m3: additive margin softmax (*AM-Softmax*)

# Additive margin softmax

- It aims to involve a margin factor *m* to enlarge the margin between target logits and non-target logits.

$$\mathcal{L}_{\text{AM-Softmax}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s(\cos(\theta_{y_i,i})-m)}}{e^{s(\cos(\theta_{y_i,i})-m)} + \sum_{j\neq y_i} e^{s(\cos(\theta_{j,i}))}}$$

- Intuitively, it will pay more attention on target logits than non-target logits, and separates target and non-target classes.

?

# *m* does not boost margin

$$\mathcal{L}_{\text{AM-Softmax}} = \frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s(\cos(\theta_{y_i,i})-m)}+\sum_{j\neq y_i}e^{s(\cos(\theta_{j,i}))}}{e^{s(\cos(\theta_{y_i,i})-m)}}$$

$$= \frac{1}{N}\sum_{i=1}^{N}\log\{1+\sum_{j\neq y_i}e^{-s(\cos(\theta_{y_i,i})-\cos(\theta_{j,i})-m)}\}$$

$$= \frac{1}{N}\sum_{i=1}^{N}\log\{1+e^{sm}\sum_{j\neq y_i}e^{-s(\cos(\theta_{y_i,i})-\cos(\theta_{j,i}))}\}$$

- Setting $s$ = 1 and $m$ = 0, it recovers the modified Softmax…
- *m only* changes the loss landscape, but *not* enlarges the margin between the target and non-target logits.

# For easy samples

$$\frac{e^{(\cos(\theta_{y_i,i})-m)}}{e^{(\cos(\theta_{y_i,i})-m)} + \sum_{j\neq y_i} e^{(\cos(\theta_{j,i}))}} \approx 1$$

$$\log\{1 + e^m \sum_{j\neq y_i} e^{-(\cos(\theta_{y_i,i})-\cos(\theta_{j,i}))}\}$$
$$\approx \boxed{e^m} \sum_{j\neq y_i} e^{-(\cos(\theta_{y_i,i})-\cos(\theta_{j,i}))}$$

- When *m* increases from 0, the contribution of easy samples will be emphasized.

# For hard samples

$$\frac{e^{(\cos(\theta_{y_i,i})-m)}}{e^{(\cos(\theta_{y_i,i})-m)}+\sum_{j\neq y_i} e^{s(\cos(\theta_{j,i}))}} \ll 1$$

$$\log\{1+e^{m}\sum_{j\neq y_i} e^{-(\cos(\theta_{y_i,i})-\cos(\theta_{j,i}))}\}$$
$$\approx \quad \boxed{m+}\log\sum_{j\neq y_i} e^{-(\cos(\theta_{y_i,i})-\cos(\theta_{j,i}))}$$

- Setting any *m* will not change the optimum.

# A brief summary

- Setting a large *m* can boost the contribution of easy samples, while is invalid to hard samples.
- This is more like a center loss which shrinks intra-speaker distribution rather than a true margin loss.
- This is not a good property as hard samples are always more concerning !
- This may overfit to easy samples and lead to bad generalization capability (inferior performance on open-set ASV).

# *Real* additive margin softmax

- AM-Softmax

$$\mathcal{L}_{\text{AM-Softmax}} = \frac{1}{N}\sum_{i=1}^{N}\log\{1+\sum_{j\neq y_i} e^{-s(\cos(\theta_{y_i,i})-\cos(\theta_{j,i})-m)}\}$$

- Max-margin training

$$\mathcal{L}_{\text{margin}} = \boxed{\max}(0, d_p - d_n + m)$$

- Real AM-Softmax

$$\mathcal{L}_{\text{RAM-Softmax}} = \frac{1}{N}\sum_{i=1}^{N}\log\{1+\sum_{j\neq y_i} e^{\max\{0,-s(\cos(\theta_{y_i,i})-\cos(\theta_{j,i})-m)\}}\}$$

# *Real* additive margin softmax

- Real AM-Softmax

$$\mathcal{L}_{\text{RAM-Softmax}} = \frac{1}{N}\sum_{i=1}^{N}\log\{1+\sum_{j\neq y_i} e^{\max\{0,-s(\cos(\theta_{y_i,i})-\cos(\theta_{j,i})-m)\}}\}$$

- If the target logit is larger than non-target logits by more than m, the exponential term will be zero, otherwise a positive loss will be incurred.
- This will encourage the model to focus on hard non-target logits, and forget easy non-targets that have been well separated.

# *Real* additive margin softmax

- Real AM-Softmax

$$\mathcal{L}_{\text{RAM-Softmax}} = \frac{1}{N}\sum_{i=1}^{N}\log\{1+\sum_{j\neq y_i} e^{\max\{0,-s(\cos(\theta_{y_i,i})-\cos(\theta_{j,i})-m)\}}\}$$

- This can will balance the contribution of all classes, which arguably alleviates the discrepancy between softmax training and the open-set ASV task.
- This can be regarded as a graft of softmax training and metric learning.

# Experiments

- Data
  - VoxCeleb: VoxCeleb2.dev, VoxCeleb1, VoxCeleb1-H/E
  - SITW: SITW.Dev.Core, SITW.Eval.Core
  - CNCeleb: CNCeleb.Train, CNCeleb.Eval
- Setting
  - X-vector architecture
  - ResNet34 topology
  - Temporal statistical pooling strategy

# Results on VoxCeleb1 and SITW

**Table 1**. EER(%) results on VoxCeleb1 and SITW.

| Objective | Hyperparameters | VoxCeleb1 | VoxCeleb1-H | VoxCeleb1-E | SITW.Dev.Core | SITW.Eval.Core |
|---|---|---|---|---|---|---|
| AM-Softmax | m = 0.20, s = 30 | **1.739** | 2.895 | 1.724 | **2.811** | 3.362 |
| Real AM-Softmax | m = 0.20, s = 30 | 1.872 | 3.068 | 1.883 | 3.466 | 3.718 |
| | m = 0.25, s = 30 | 1.819 | 2.914 | 1.781 | 3.350 | 3.554 |
| | m = 0.30, s = 30 | 1.755 | **2.812** | **1.696** | 3.003 | 3.417 |
| | m = 0.35, s = 30 | 1.808 | 2.888 | 1.747 | 2.849 | **3.335** |

- m was chosen according to the development sets.
- This improvement is not very remarkable but consistent, demonstrating that the real margin is a correct modification.

# Results on ‘Hard trials’

**Table 2**. EER(%) results on ‘hard trials’ selected from VoxCeleb and SITW with two objective functions.

| Objective | Hyperparameters | VoxCeleb1-H.H | VoxCeleb1-E.H | SITW.Eval.Core.H |
|---|---|---|---|---|
| AM-Softmax | m = 0.20, s = 30 | 39.794 | 38.970 | 36.082 |
| ARM-Softmax | m = 0.20, s = 30 | 40.729 | 40.416 | 40.206 |
| | m = 0.25, s = 30 | 39.899 | 37.814 | 35.052 |
| | m = 0.30, s = 30 | **39.175** | 36.861 | 36.082 |
| | m = 0.35, s = 30 | 39.794 | **36.821** | **32.990** |

- RAM-Softmax is significantly superior on ‘hard trials’.
- This indicates that RAM-Softmax is more robust under more challenging test conditions.

# Results on CNCeleb

**Table 2**. EER(%) results on CNCeleb.

| Objective | Hyperparameters | CNCeleb.Eval |
|---|---|---|
| AM-Softmax | m = 0.10, s = 30 | 11.450 |
| Real AM-Softmax | m = 0.10, s = 30 | 11.618 |
| | m = 0.15, s = 30 | 11.323 |
| | m = 0.20, s = 30 | **11.049** |
| | m = 0.25, s = 30 | 11.422 |

- Again, RAM-Softmax outperforms AM-Softmax on this more challenging dataset.

# Conclusions

- We analyze that AM-Softmax loss cannot conduct real margin training. It is more like a center loss rather than a true margin loss.
- RAM-Softmax is a graft of angular softmax training and max-margin metric learning, and can improve the generalization capability on open-set tasks.
- RAM-Softmax obtains marginal but consistent performance improvement on normal test conditions, while obtains notably performance improvement on complicated test conditions.