



Large Margin Softmax Loss for Speaker Verification

Yi Liu, Liang He, Jia Liu

Tsinghua National Laboratory for Information Science and Technology,
Department of Electronic Engineering, Tsinghua University, Beijing 100084, China

liu-yi15@mails.tsinghua.edu.cn, {heliang, liuj}@tsinghua.edu.cn

Abstract

In neural network based speaker verification, speaker embedding is expected to be discriminative between speakers while the intra-speaker distance should remain small. A variety of loss functions have been proposed to achieve this goal. In this paper, we investigate the large margin softmax loss with different configurations in speaker verification. Ring loss and minimum hyperspherical energy criterion are introduced to further improve the performance. Results on VoxCeleb show that our best system outperforms the baseline approach by 15% in EER, and by 13%, 33% in minDCF08 and minDCF10, respectively.

Index Terms: speaker verification, speaker embedding, large margin softmax, Ring loss, minimum hyperspherical energy

1. Introduction

Recently, speaker embedding, which is extracted from a deep neural network, outperforms the conventional i-vector in many speaker verification tasks [1]. By virtue of the excellent performance, speaker embedding is becoming the next generation of speaker recognition technology. Similar with i-vector, speaker embedding encodes a variable-length utterance into a fix-length vector representing the speaker characteristics. A variety of backend classifiers can be applied to suppress noise and session variability. Speaker embedding can also be used in other applications such as speaker diarization [2], speaker retrieval and speech synthesis.

Speaker verification is an open-set recognition problem. An utterance is verified to be a certain speaker if their similarity exceeds a threshold. Ideal speaker embedding should be discriminative between different speakers and compact within the same speaker. Although cross-entropy with softmax is arguably one of the most commonly used loss function to train the speaker embedding neural network, it is more suitable for classification and does not explicitly encourage discriminative learning of features.

To address this issue, different loss functions are proposed. Triplet loss for speaker verification were first presented in [3, 4]. By selecting appropriate training samples, triplet loss performed well in both text-dependent and text-independent tasks. However, the performance is sensitive to the triplet mining strategy [5, 6] and it is time consuming to design such a training procedure. Also, speaker identity subspace loss [7], Gaussian Mixture loss [8], etc., were proposed in other works.

On the other hand, efforts have been made to improve the original softmax loss. Center loss was introduced in [9, 10] to constrain features to be gathered around the corresponding centers and thus reduce the intra-speaker variability. Both triplet loss and center loss are optimized in the Euclidean space. In the last few years, angular-based losses have become popular. Compare with the Euclidean distance, angular distance is a more natural choice in the feature space. In [11, 12, 13], the

features and the weights of the output layer were normalized before softmax, making the loss function focus on the cosine similarity. Generalized end-to-end loss was proposed in [14]. The scaled cosine scores between the features and the estimated speaker centers were used as the logits to compute the loss. Angular margin softmax loss was first presented in [15, 16] in which the margin is incorporated with the angle in a multiplicative way. This method is extended in [17, 18, 19] where additive margins are used. Some of these losses have been applied in speaker verification [20, 21, 22, 23, 24, 25]. Since all these losses combine softmax with margins, we call them the large margin softmax loss in this paper.

In this paper, we first build a baseline system using a generic toolkit similar with [26]. Several training strategies are used to improve the accuracy. We then compare the performance of the large margin softmax loss using different configurations. Ring loss [27] and minimum hyperspherical energy (MHE) criterion [28] are involved to enhance the discriminative learning and enlarge the inter-speaker separability. Experiments on VoxCeleb show that our baseline system achieves better performance than the Kaldi x-vector recipe and reduces the EER, minDCF08 and minDCF10 from 3.10%, 0.0169, 0.4977 to 2.34%, 0.0122 and 0.3754, respectively. Using the large margin softmax loss with auxiliary objective functions, the best system further improves the performance to 2%, 0.0106 and 0.2487.

The organization of this paper is as follows. The speaker embedding we use is briefly introduced in Section 2. Section 3 describes the large margin softmax loss and different techniques to enhance the loss. Our experimental setup and results are given in Section 4 and 5. The last section concludes the paper.

2. Speaker embedding

The deep neural network used to extract speaker embedding consists of frame-level and segment-level sub-networks, connected by a temporal pooling layer. The frame-level network can be seen as a speaker feature extractor which transforms the acoustical features into speaker-related vectors. These vectors are aggregated across the entire utterance by a pooling layer and further processed by several fully-connected layers. Different loss functions can be used to optimize the network. After training, the output of a hidden layer at the segment-level network is extracted as the speaker embedding. Cosine scoring, LDA and PLDA are usually applied to generate the verification scores. Since the output layer is removed during the test phase, the test speakers do not have to be present in the training data.

In this paper, speaker embedding is extracted from the x-vector architecture [1]. The x-vector is popular on many applications and has been provided as the official system on the recent NIST speaker recognition evaluation (SRE). The details are described in Section 4.2.

3. Large margin softmax loss

3.1. Definition

The widely used softmax loss is presented as

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

where N is the batch size, C is the number of speakers in the training set, \mathbf{w}_j is the j -th column of the weights in the output layer. \mathbf{x}_i is the input of the last (i.e. output) layer, y_i is the ground truth label for the i -th sample. To avoid ambiguity, we use *feature* to represent \mathbf{x}_i in this paper while *embedding* denotes the speaker embedding extracted from a hidden layer of the network. The logit $\mathbf{w}_j^T \mathbf{x}_i$ can be transformed to $\|\mathbf{w}_j\| \|\mathbf{x}_i\| \cos \theta_j$, where θ_j is the angle between \mathbf{w}_j and \mathbf{x}_i . This indicates that Eq. 1 is not only influenced by the angle θ_j between the feature and the weights but also the norm \mathbf{w}_j and \mathbf{x}_i . In modified softmax [16], the weights are L_2 normalized as $\|\mathbf{w}_j\| = 1$ and the loss is

$$L_{MS} = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{s \cos \theta_{y_i}}}{\sum_{j=1}^C e^{s \cos \theta_j}} \quad (2)$$

where s is a scaling factor. This factor can be the feature norm $\|\mathbf{x}_i\|$ or a fixed value if the feature is also normalized [12]. We will discuss the impact of the feature normalization later.

Based on Eq. 2, different margins can be introduced by reformulating the target logit. We define an angle function [19]

$$\psi(\theta_{y_i}) = \cos(m_1 \theta_{y_i} + m_2) - m_3 \quad (3)$$

as the target logit and rewrite Eq. 2 as the large margin softmax loss

$$L_{LMS} = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{s \cdot \psi(\theta_{y_i})}}{e^{s \cdot \psi(\theta_{y_i})} + \sum_{j=1, j \neq i}^C e^{s \cdot \cos \theta_j}} \quad (4)$$

where m_1, m_2 and m_3 are margins. Strictly, Eq. 3 is only valid when $m_1 = 1$ and $m_2 = 0$, since $\forall \theta_{y_i} \in [0, \pi]$, $\psi(\theta_{y_i})$ should be a monotonically decreasing function. However, in practice, the angle is usually in the range of $[0.35, 1.75]$ [19]. We can safely apply Eq. 3 and 4 to optimize the network when $m_1 \approx 1$ and $m_2 < 1$. When m_1 is large, Eq. 3 becomes invalid and a new $\psi(\theta_{y_i})$ is required. Let $m_2 = 0, m_3 = 0$. For $m_1 \geq 2$, we use the angle function defined in [16]

$$\psi(\theta_{y_i}) = (-1)^k \cos(m_1 \theta_{y_i}) - 2k \quad (5)$$

where $\theta_{y_i} \in [\frac{k\pi}{m_1}, \frac{(k+1)\pi}{m_1}]$ and $k \in [0, m_1 - 1]$. The curves of the angle functions using different margins are illustrated in Fig. 1.

The margins m_1, m_2 and m_3 can be used separately [16, 18, 17, 19]. Also, they can be further combined as in [19]. In this paper, we only use one single margin at a time since the performance gain of the combination is relatively small while large efforts will be paid to tune the hyperparameters. We denote the losses as angular softmax (*ASoftmax*), additive angular margin softmax (*ArcSoftmax*) and additive margin softmax loss (*AMSoftmax*) when m_1, m_2 and m_3 are used, respectively.

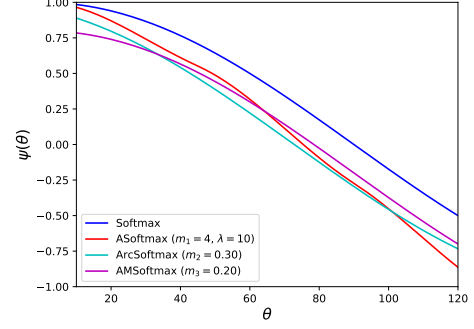


Figure 1: The angle function $\psi(\theta)$ using different margins. The configurations are selected as they performs the best in our experiments.

3.2. Feature normalization

As discussed in [11], the L_2 norm of the feature is related with the sample quality when the softmax loss is used. The network will minimize the loss by simply increasing the norm of the features for easy samples while keeping the norm of the hard samples small. This avoids the network processing samples in poor quality well.

To solve this issue, feature normalization is presented in many works. After normalization, the feature norm is eliminated from the loss and a fixed-value scaling factor s is used instead. Using the feature normalization, the loss is only related with the angle function. Features with small norm will get much bigger gradients compared to those with large norm, making the network pay more attention to the low-quality samples [17].

Feature normalization uses an additional layer to apply the length normalization rather than learning to map the features into a fixed-norm hypersphere. In this paper, however, we introduce Ring loss [27] to directly apply the norm constraint on the features. The definition of the Ring loss is straightforward. We want the feature norm to be close to a target value R . An auxiliary loss is employed as

$$L_R = \frac{\lambda_R}{N} \sum_{i=1}^N (\|\mathbf{x}_i\| - R)^2 \quad (6)$$

where λ_R is the loss weight with the primary large margin softmax loss. The Ring loss can be considered as a *soft* version of feature normalization and the target norm R can be learnt during the network training.

3.3. Enlarge inter-speaker feature separability

Although the large margin softmax loss improves the intra-class compactness, it does not explicitly promote the inter-class separability. In [28], the authors proposed a minimum hyperspherical energy (MHE) criterion to encourage the weights of the output layer to distribute evenly on hypersphere. The MHE criterion is expressed as

$$L_M = \frac{\lambda_M}{N(C-1)} \sum_{i=1}^N \sum_{j=1, j \neq i}^C f(\|\hat{\mathbf{w}}_{y_i} - \hat{\mathbf{w}}_j\|) \quad (7)$$

where λ_M is a weighting hyperparameter, $\hat{\mathbf{w}}_{y_i}, \hat{\mathbf{w}}_j$ are the L_2 normalized weights in the loss function and $f(z) = 1/z^2$ is a decreasing function. Intuitively, MHE loss enlarges the overall inter-class feature separability. Similar with the Ring loss, we include MHE as an auxiliary objective function.

3.4. Annealing strategy during training

From a classification perspective, the large margin softmax makes the decision boundary more stringent to correctly classify \mathbf{x}_i . From a view of optimization, the existence of the margin makes the well-separated features continue to get big gradients which can shrink the intra-class variance.

However, as a side effect, the margin will also increase the training difficulty especially when the network is randomly initialized. To stable the training procedure, an annealing strategy is applied. The target logit is replaced by the weighted average of the original logit and the large margin counterpart, which means

$$\psi_{\text{train}}(\theta_{y_i}) = \frac{1}{1+\lambda}\psi(\theta_{y_i}) + \frac{\lambda}{1+\lambda}\cos(\theta_{y_i}) \quad (8)$$

where $\lambda = \max\{\lambda_0, \lambda_b \cdot (1 + \gamma \cdot s)^{-\alpha}\}$, s is the training step, λ_0 is the minimum value it can achieve, λ_b , γ and α are the hyperparameters controlling the annealing speed. The annealing strategy lessens the impact of the margin and makes the network converge more quickly in the beginning of the training.

3.5. Other discussions

In [14], generalized end-to-end (GE2E) loss is proposed to train the speaker network in a large-scale dataset. We rewrite Eq. 6 in [14] as

$$L_{GE2E} = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{s \cdot \cos(\mathbf{x}_i, \mathbf{c}_{y_i}) + b}}{\sum_{j=1}^{C'} e^{s \cdot \cos(\mathbf{x}_i, \mathbf{c}_j) + b}} \quad (9)$$

where C' is the number of speakers in a minibatch and \mathbf{c}_j is the center of speaker j estimated from the batch. The softmax is computed across the batch rather than the entire dataset. This is convenient when the training set is extremely large. If the bias b is omitted and the estimated center \mathbf{c}_j is replaced with a learnable weight \mathbf{w}_j , the GE2E loss becomes the modified softmax in Eq. 2, which can be further incorporated with margins. Hence, combining with the GE2E loss, the large margin softmax loss has potential to be applied on a dataset comprising millions of speakers.

4. Experimental setup

4.1. Dataset

To investigate the performance of the large margin softmax loss, we have run experiments on the VoxCeleb dataset [29, 30]. The training set includes VoxCeleb1 *dev* set and VoxCeleb2. The VoxCeleb1 *test* set is used as the evaluation set. This setup is selected to be consistent with the Kaldi recipe [31].

Equal error rate (EER), minimum detection cost function from NIST SRE08 (minDCF08) [32] and SRE10 (minDCF10) [33] are presented to demonstrate the performance.

4.2. Training details

The acoustic feature in our experiments is 30-dim MFCCs with cepstral mean normalization. An energy-based voice active detection (VAD) is applied. The training data is augmented using MUSAN [34] and RIR [35].

We use the same network architecture as Kaldi [1] to extract x-vectors with the following modifications.

- For the frame-level network, a 5-layer TDNN is used. The kernel size for each layer is [5, 5, 7, 1, 1]. Different with Kaldi,

there is no dilation used. This performs better in our experiments and is also suggested in other works [26]. Statistics pooling and a 2-layer segment-level network is appended after the frame-level network.

- Each hidden layer consists of an affine component followed by batch-normalization (BN) and ReLU non-linearity. The order of BN and ReLU does not necessarily lead to better performance but the training is more stable than that of the opposite order.
- The last ReLU in the segment-level network is removed. The non-linearity limits the feasible angles between the feature and the weights which is not a good choice when the angle-based large margin softmax loss is applied [16].

At every training step, we sample 64 speakers. For each speaker, a segment with 200 to 400 frames is sliced from the utterances. Softmax with cross entropy is used to train the baseline system. L_2 regularization is applied to all layers in the network to prevent overfitting. We select stochastic gradient descent (SGD) as the optimizer and the initial learning rate is set to 0.01. A 1000-utterance validation set is randomly selected from the training set and the learning rate is halved if the validation loss gets stuck for a while. The loss converges after the learning rate goes down below 10^{-5} , resulting to around 2.5M training steps. No dropout is applied in our networks as described in [26].

When training with the large margin softmax loss, a λ annealing strategy is used as described in Section 3.4. Specifically, we set a fast decay for AMSOftmax where $\lambda_b = 1000$, $\gamma = 0.0001$, $\alpha = 5$. For ASOftmax and ArcSOftmax, the decay speed is slowed down by setting $\gamma = 0.00001$. The λ_0 for ArcSOftmax and AMSOftmax is 0, while for ASOftmax, $\lambda_0 = 10$. This non-zero λ results in a more gentle angle function. For example, when $m_1 = 4$, $\lambda = 10$, the target logit is similar to ASOftmax with $m_1 = 1.2$ without λ .

After training, the output of the second last layer in the segment-level network is extracted as the speaker embedding. LDA is used to reduce the dimension to 200 and PLDA is then applied to generate the verification scores. One may also use the embedding extracted from the last layer with simple cosine backend. However, in our experiments, we find the PLDA scoring generally performs better.

Our systems are implemented with Kaldi and Tensorflow toolkits. The code and models have been released ¹.

5. Results

Table 1 summarizes the results of different systems. The first row in Table 1 shows the Kaldi recipe for VoxCeleb. The first experiment is to validate the performance of our baseline system. We find a large weight decay parameter works well in our systems. When increasing this parameter from 0.0001 to 0.01, the EER is improved from 3% to 2.34%. The second row shows the performance of our baseline system which is significantly outperform the standard Kaldi result. The third row is the result using the modified softmax loss. Without any margins, the modified softmax does not perform better by simply normalizing the weights.

The performance of the large margin softmax loss is exhibited in the following sections of Table 1. We remove the last ReLU for all these networks. This generally improves the results. For instance, the performance of ASOftmax ($m_1=4$) with

¹<https://github.com/mycrazycracy/tf-kaldi-speaker>

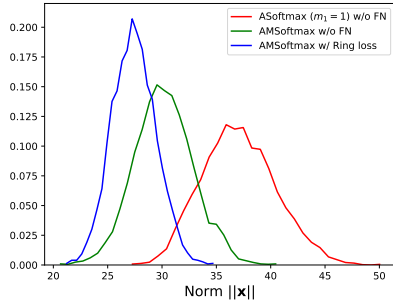


Figure 2: The norm distribution on the test set. When the Ring loss is used, the norm variance becomes smaller.

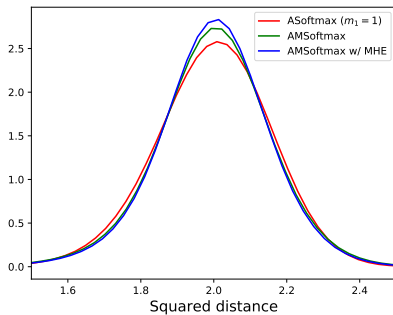


Figure 3: The distribution of the squared distances between the normalized weights in the output layer. The means for all the systems is about 2.0 and the MHE loss reduces the variance.

ReLU is 2.12%, 0.0122 and 0.3214 in EER, minDCF08 and minDCF10, while without ReLU, it achieves 2.15%, 0.0113 and 0.3108 instead. The same trends are observed in other systems as well.

From Table 1, it is clear that ASoftmax achieves the best result when $m_1 = 4$. The performance of ArcSoftmax is similar with ASoftmax and the best margin is about $m_2 = 0.25$ to $m_2 = 0.30$. The AMSoftmax performs the best among all these large margin softmax losses with the optimal margin $m_3 = 0.20$. We notice that the best margins for these systems are relatively small compared with those reported in the face verification [16, 17, 19].

We now investigate the influence of the Ring loss and the MHE loss. The weight λ_R for the Ring loss is set as 0.01 and R is initialized at 20. Table 1 shows that the Ring loss improves the minDCF08 and minDCF10. The norm distributions of different systems are presented in Fig. 2. Since the weights of the softmax network are not normalized, the mean of the feature norm is very large (about 150). To show the feature norm without margins, we use the modified softmax instead. From Fig. 2, we find that using the margin helps to reduce the norm variance. The margin prevents the norm of the simple samples growing too large. The norm distribution further shrinks when the Ring loss is applied. However, even though the network is trained using AMSoftmax without feature normalization, the norm variance is relatively small. Therefore, the effectiveness of the Ring loss with AMSoftmax is less significant in our experiment.

The performance of the MHE loss is presented in the last row of Table 1. The weight λ_M is 0.01. The AMSoftmax with MHE loss achieves the best result among all the systems. This

Table 1: The comparison of results between systems training with different loss functions. The weights for the Ring loss and the MHE loss are both 0.01.

		EER(%)	minDCF08	minDCF10
Kaldi ²		3.10	0.0169	0.4977
Softmax		2.34	0.0122	0.3754
Modified Softmax		2.62	0.0131	0.4146
ASoftmax	$m_1 = 2$	2.18	0.0119	0.3791
	$m_1 = 4$	2.15	0.0113	0.3108
ArcSoftmax	$m_2 = 0.20$	2.14	0.0119	0.3610
	$m_2 = 0.25$	2.03	0.0120	0.4010
	$m_2 = 0.30$	2.12	0.0115	0.3138
	$m_2 = 0.35$	2.23	0.0123	0.3622
AMSoftmax	$m_3 = 0.15$	2.13	0.0113	0.3707
	$m_3 = 0.20$	2.04	0.0111	0.2922
	$m_3 = 0.25$	2.15	0.0119	0.3559
	$m_3 = 0.30$	2.18	0.0115	0.3152
AMSoftmax + Ring Loss		2.07	0.0107	0.2687
AMSoftmax + MHE		2.00	0.0106	0.2487

loss improves the baseline performance by 15% in EER, 13% in minDCF08 and 33% in minDCF10. To get some insights of the MHE loss, we illustrate the distribution of the pairwise squared distances between the normalized weights. The distance, which is $\|\hat{w}_i - \hat{w}_j\|^2, \forall i, j, i \neq j$, indicates the separability between speakers on the training set. In Fig. 3, it is shown that all the distributions have the means at about 2.0, indicating $\cos(\hat{w}_i, \hat{w}_j)$ is 0 in average. The AMSoftmax with the MHE loss achieves the smallest variance of the inter-speaker distances, which means the features of speakers distribute more evenly on hypersphere, leading to a better overall separability in the feature space.

6. Conclusions

In this paper, we investigate the large margin softmax loss for speaker verification. By selecting an appropriate margin, the large margin softmax loss can achieve promising results. Ring loss and MHE loss are involved to further improve the performance. Ring loss is a soft version of feature normalization and alleviates the impact of feature norm. MHE criterion is another loss function which enlarges the overall inter-speaker separability. On VoxCeleb, our baseline system achieves better result than the Kaldi toolkit. We find AMSoftmax is easier to train and generally performs better than ASoftmax and ArcSoftmax in our experiments. The best system is obtained when AMSoftmax is used with the MHE loss. This combination substantially outperforms the baseline.

In the future, more efforts will be made to enable simple cosine scoring and remove the need for the PLDA backend.

7. Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant No. 61403224 and No. U1836219.

²<https://github.com/kaldi-asr/kaldi/blob/master/egs/voxceleb/v2>

8. References

- [1] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, “X-vectors: Robust dnn embeddings for speaker recognition,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5329–5333.
- [2] D. Snyder, D. Garcia-Romero, G. Sell, A. McCree, D. Povey, and S. Khudanpur, “Speaker recognition for multi-speaker conversations using x-vectors,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.
- [3] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, and Z. Zhu, “Deep speaker: an end-to-end neural speaker embedding system,” *arXiv preprint arXiv:1705.02304*, 2017.
- [4] C. Zhang and K. Koishida, “End-to-end text-independent speaker verification with triplet loss on short utterances,” in *Proc. Interspeech*, 2017, pp. 1487–1491.
- [5] A. Hermans, L. Beyer, and B. Leibe, “In defense of the triplet loss for person re-identification,” *arXiv preprint arXiv:1703.07737*, 2017.
- [6] C.-Y. Wu, R. Manmatha, A. J. Smola, and P. Krahenbuhl, “Sampling matters in deep embedding learning,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2840–2848.
- [7] R. Ji, X. Cai, and B. Xu, “An end-to-end text-independent speaker identification system on short utterances,” in *Proc. Interspeech*, 2018.
- [8] W. Wan, Y. Zhong, T. Li, and J. Chen, “Rethinking feature distribution for loss functions in image classification,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9117–9126.
- [9] N. Li, D. Tuo, D. Su, Z. Li, and D. Yu, “Deep discriminative embeddings for duration robust speaker verification,” in *Proc. Interspeech*, 2018, pp. 2262–2266.
- [10] S. Yadav and A. Rai, “Learning discriminative features for speaker identification and verification,” in *Proc. Interspeech*, 2018, pp. 2237–2241.
- [11] R. Ranjan, C. D. Castillo, and R. Chellappa, “L2-constrained softmax loss for discriminative face verification,” *arXiv preprint arXiv:1703.09507*, 2017.
- [12] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille, “Normface: 12 hypersphere embedding for face verification,” in *Proceedings of the 25th ACM international conference on Multimedia*. ACM, 2017, pp. 1041–1049.
- [13] C. Zhang, F. Bahmaninezhad, S. Ranjan, H. Dubey, W. Xia, and J. H. Hansen, “UTD-CRSS systems for 2018 NIST speaker recognition evaluation,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 5776–5780.
- [14] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, “Generalized end-to-end loss for speaker verification,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4879–4883.
- [15] W. Liu, Y. Wen, Z. Yu, and M. Yang, “Large-margin softmax loss for convolutional neural networks,” in *Proc. ICML*, 2016, pp. 507–516.
- [16] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, “Sphreface: Deep hypersphere embedding for face recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 212–220.
- [17] F. Wang, J. Cheng, W. Liu, and H. Liu, “Additive margin softmax for face verification,” *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 926–930, 2018.
- [18] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, “Cosface: Large margin cosine loss for deep face recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5265–5274.
- [19] J. Deng, J. Guo, and S. Zafeiriou, “Arcface: Additive angular margin loss for deep face recognition,” *arXiv preprint arXiv:1801.07698*, 2018.
- [20] Z. Huang, S. Wang, and K. Yu, “Angular softmax for short-duration text-independent speaker verification,” in *Proc. Interspeech*, 2018, pp. 3623–3627.
- [21] Y. Li, F. Gao, Z. Ou, and J. Sun, “Angular softmax loss for end-to-end speaker verification,” *arXiv preprint arXiv:1806.03464*, 2018.
- [22] G. Bhattacharya, J. Alam, and P. Kenny, “Adapting end-to-end neural speaker verification to new languages and recording conditions with adversarial training,” *arXiv preprint arXiv:1811.03055*, 2018.
- [23] M. Hajibabaei and D. Dai, “Unified hypersphere embedding for speaker recognition,” *arXiv preprint arXiv:1807.08312*, 2018.
- [24] W. Xie, A. Nagrani, J. S. Chung, and A. Zisserman, “Utterance-level aggregation for speaker recognition in the wild,” *arXiv preprint arXiv:1902.10107*, 2019.
- [25] R. Li, N. Li, D. Tuo, M. Yu, D. Su, and D. Yu, “Boundary discriminative large margin cosine loss for text-independent speaker verification,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6321–6325.
- [26] H. Zeinali, L. Burget, J. Rohdin, T. Stafylakis, and J. Cernocky, “How to improve your speaker embeddings extractor in generic toolkits,” *arXiv preprint arXiv:1811.02066*, 2018.
- [27] Y. Zheng, D. K. Pal, and M. Savvides, “Ring loss: Convex feature normalization for face recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5089–5097.
- [28] W. Liu, R. Lin, Z. Liu, L. Liu, Z. Yu, B. Dai, and L. Song, “Learning towards minimum hyperspherical energy,” in *Advances in Neural Information Processing Systems*, 2018, pp. 6225–6236.
- [29] A. Nagrani, J. S. Chung, and A. Zisserman, “Voxceleb: a large-scale speaker identification dataset,” in *Proc. Interspeech*, 2017, pp. 2616–2620.
- [30] J. S. Chung, A. Nagrani, and A. Zisserman, “Voxceleb2: Deep speaker recognition,” in *Proc. Interspeech*, 2018, pp. 1086–1090.
- [31] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, and Others, “The kaldi speech recognition toolkit,” in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2011.
- [32] A. F. Martin and C. S. Greenberg, “NIST 2008 speaker recognition evaluation: Performance across telephone and room microphone channels,” in *Proc. Interspeech*, 2009, pp. 2579–2582.
- [33] —, “The NIST 2010 speaker recognition evaluation,” in *Proc. Interspeech*, 2010, pp. 2726–2729.
- [34] D. Snyder, G. Chen, and D. Povey, “Musan: A music, speech, and noise corpus,” *arXiv preprint arXiv:1510.08484*, 2015.
- [35] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, “A study on data augmentation of reverberant speech for robust speech recognition,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 5220–5224.