ANGLEVOICE: LEVERAGING ANGULAR MARGIN LOSSES FOR REAL WORLD SPEAKER RECOGNITION, CLUSTERING AND DIARIZATION

Daniel Neururer [1], *Claude Lehmann* [1], *Patrick Walter, Jan Sonderegger, Thilo Stadelmann* [1]

[1] ZHAW Datalab
Zurich University of Applied Sciences
Winterthur, Switzerland

neurudan@students.zhaw.ch, lehmacl1@students.zhaw.ch,
patrick.walter214@gmail.com, <jan email>, stdm@zhaw.ch

ABSTRACT

This paper is based on the ZHAW Deep Voice project, which consists of several speaker clustering models trained on TIMIT [1] and VoxCeleb2 [2]. Recent work [3] led to the conclusion that the Kullback-Leibler Divergence Loss does not scale well, when the number of labels in the training set is increased. Angular margin losses like ArcFace [4], CosFace [5] and SphereFace [6] do not have this problem and teach a model to separate labels better, by adding a margin in the angular or cosine space to the true label of a prediction and yielded promising results in face recognition tasks. We anticipate those losses to suit speaker clustering tasks as well. To prove this, we have conducted several experiments and summarized them in this article.

Index Terms— Deep Learning, Speaker Recognition, Angular Margin Loss

1. INTRODUCTION

Speaker recognition is a broad research field, with many different subareas, the most difficult being speaker clustering. The goal here is to compare two utterances and decide, if they are coming from the same speaker (who is not known to a model in advance) or not.

A common approach to solve the problem is by training a Deep Neural Network (DNN) to classify the speaker of speech segments. The fully trained model is then used to extract features, also called embeddings, by retrieving the output of the model after a certain layer. Those are then fed to a clustering algorithm, which generates clusters of speakers.

In previous experiments conducted at ZHAW Datalab [7][8][3], networks have been trained on TIMIT [1] using the Pairwise Kullback-Leibler Divergence (PKLD) loss function. Altough those models have outperformed state-of-the-art results on a training set with 100 speakers, recent work of Sonderegger and Walter [3] showed that when using PKLD on a larger training set consisting of 470 speakers, the performance drops significantly. Therefore, the project was in need of a better matching loss function to overcome the problem.

We considered angular margin losses like CosFace [5], ArcFace [4] and SphereFace [6] as suitable losses, since they have the benefit of boosting inter-class diversity and intra-class compactness. Furthermore, they use an increased amount of training classes to their advantage.

Albeit those losses being designed for face recognition applications, we were able to prove that they also can be applied in speaker clustering. Despite our approach only being close to the state-of-the-art when trained using 100 speakers, it exceeds all the previous results when trained on 470 speakers.

2. RELATED WORK

Angular margin losses have already been introduced in speaker recognition applications by Xie et al. [9]. They applied the ArcFace loss on a thin-ResNet structure, combined with a dictionary-based NetVLAD or GhostVLAD layer to aggregate features across time, and achieved state-of-the-art results on the VoxCeleb1 and VoxCeleb2 datasets.

2.1. ZHAW Deep Voice

The ZHAW Deep Voice project is the result of several bachelor theses that were conducted at the Institute of Applied Information Technologies at the Zurich University of Applied Sciences. Its purpose is the research of speaker clustering methods and consists of a CNN and a LSTM approach, both of them following the approach mentioned in the Introduction. The clustering algorithm used for all the conducted experiments is a hierarchical agglomerative clustering with complete linkage and the cosine metric.

The CNN approach originally used a common categorical crossentropy loss to train the model, which later has been modified by using the PKLD loss. This newly introduced loss

is comparing embeddings pairwise. Its intention is to produce embeddings that are similar to embeddings of the same speaker, but dissimilar to those of different speakers. The similarity between pairs is calculated by using their Kullback-Leibler Divergence.

Since speech data is highly time dependent, CNNs are by design not as well suited for the problem as RNNs. Therefore, a second model using bidirectional LSTM layers, that also makes use of the PKLD loss, has been implemented.

Latest results [3] led to the conclusion that when using PKLD, the probability of a segment being compared to another one having the same speaker decreases, as we increase the number of speakers in the training set. While the network takes its focus on inter-class diversity, it nearly ignores intra-class compactness, resulting in a massive performance drop.

Until recently, the project has only been trained and evaluated on the TIMIT dataset. Since this dataset is small and was recorded in studio conditions, it is not so expressive, but more importantly, it does not represent real world conditions. In the work of Lehmann and Lauener [10], the VoxCeleb2 dataset has been introduced. It contains over a million utterances of 6112 speakers that have been extracted from video clips uploaded to YouTube. However, due to the change of environment and increasement of the number of speakers, the results did not meet our goals.

2.2. Angular Margin Losses

In Face Recognition, there has been a need for a loss function that enhances the discriminative power of the network for large scale datasets. Possible solutions for the problem are CosFace [5], ArcFace [4] and SphereFace [6]. They are an extension of the widely used softmax loss, which is defined in the following equation:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}} \tag{1}$$

Where N and n denote the batch size, number of classes. y_i is the ground truth class of the i-th sample.

We recall that the softmax loss sees the logits $W_j^T x_i$ as a linear combination of the features x_i and the weights W_j of class j at the last fully connected layer. The three losses all see W_j as the class centre in an angular space, thus transform the logit to: $W_j^T x_i = \|W_j\|\|x_i\|\cos(\theta_j)$, with θ_j being the angle between W_j and x_i. To simplify the problem, the bias b is set to zero.

To be able to recieve θ_j, the features and the weights are being L2 normalized, which results in $\|W_j\| = \|x_i\| = 1$ and therefore $W_j^T x_i = \cos(\theta_j)$. Now each of the losses add a margin to the ground truth class, while leaving the rest of the logits untouched. CosFace adds a margin m_c to the logit in the cosine space, while ArcFace and SphereFace add m_a and multiply m_s a margin, respectively, in the angular space. In the end, all the resulting logits are being rescaled by a fixed feature norm s, which all together results in $cos(m_s\theta_{y_i} + m_a) - m_c$. The remaining steps are the same as in the softmax loss. All three losses can easily be combined, as visible in the next equation, which enables us to further boost performance.

$$L = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s(cos(m_s\theta_{y_i}+m_a)-m_c)}}{e^{s(cos(m_s\theta_{y_i}+m_a)-m_c)} + \sum_{j=1,j\neq y_i}^{n} e^{s\cos\theta_j}} \tag{2}$$

Unlike PKLD, those losses benefit of an increased number of classes in the training set. Furthermore, they are boosting intra-class compactness and inter-class diversity, while being computationally efficient. All three losses showed promising results when applied to face recognition. ArcFace was even able to consistently outperform the state-of-the-art.

3. EXPERIMENTS

To prove that the angular margin loss functions are a desireable substitute for the PKLD loss, several experiments have been conducted. In the first experiment, we tried to exchange the PKLD loss with the proposed angular margin losses, while not changing the rest of the network at all. In the second experiment, our goal was to research the effect on the performance of the model, when changing the dimension in the bottleneck layer. In the third experiment, the model has been changed once more to only use the proposed loss functions, but was trained and evaluated on the VoxCeleb2 dataset.

All experiments are benchmarked using the *Misclassification Rate (MR)*, *Average Cluster Purity (ACP)*, *Adjusted RAND Index (ARI)* and *Diarization Error Rate (DER)* metrics. However, this work sets its focus on the MR metric, since it shows how many utterances are not linked to the correct cluster and we want to be able to compare it to our earlier work. We use it as introduced in [11], which is defined as: MR $= \frac{1}{N}\sum_{j=1}^{N_c} e_j$. N and N_c denotes the total number of embeddings and the total number of found clusters, respectively. e_j is the amount of embeddings in cluster j, that are not mapped with the correct cluster. We define a cluster as unique and correct, if it is the one containing the largest amount of embeddings from the corresponding speaker, and if the amount of embeddings from that speaker is larger than the amounts of embeddings from other speakers.

We have used a time of 400ms for the utterances in all our setups, since earlier work [8] proved this to be a sweet spot for the clustering.

3.1. Angular Margin Loss Integration

The key idea of this experiment was to prove that we can exchange PKLD for the proposed losses, to further improve the state-of-the-art performance. The structure of the used

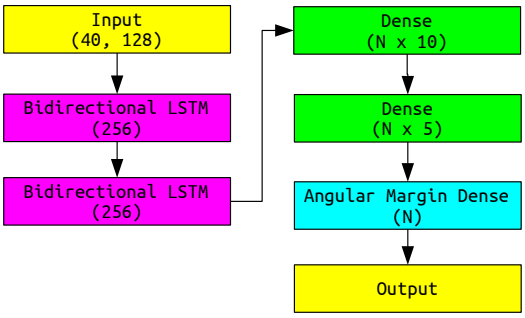

Fig. 1. Structure of the angular margin loss model, with N being the number of speakers in the training set. The Angular Margin Dense layer is a customized dense layer without a bias, that first L2-normalizes its input and its weights and then returns their dot-product.

model is visible in Figure 1. We have trained the model several times on both, the 100 and 470 speakers training set. All models have been evaluated on the 40, 60 and 80 speaker test set. For training, a parameter setting of $(m_a, m_c, m_s, s) = (0.01, 0, 1, 30)$ was used. This means, it only applies a margin of 0.01 to the ArcFace part of the loss, as well as a feature norm of 30.

Loss Function	MR (40)	MR (60)	MR (80)
PKLD (100)	2.81	3.54	5.94
Angular Margin (100)	3.21	4.76	6.79
PKLD (470)	27.50	27.92	30.63
Angular Margin (470)	**1.88**	**3.19**	**3.33**

Table 1. Averaged MR results of the PKLD and angular margin loss models in %. The number after the loss name represents the number of speakers in the training set and the number after each "MR" denotes the amount of speakers used in the test set.

When we compare the models trained on the 100 speakers set, the PKLD loss indicates a better performance than the proposed loss. When trained on the 470 speakers set on the other hand, the new model was not only able to beat PKLD, but also outperform all previous models. Thus, we proved that the angular margin loss benefits from larger amounts of speakers.

It is worth mentioning, that we consider the 40 speakers test set as critical, due to receiving several results being identical for different runs. Hence, the set seems to be too small for a secure and expressive evaluation.

Since we were unable to find a proper explanation on how to find the margin and feature norm hyperparameters of each of the three angular margin losses, we performed a gridsearch over a set of sample parameters for both training sets. We came to the conclusion that the number of speakers in the training set did influence the choosing of parameters in some ways. While both networks have the same good and bad parameters, the 470 speaker model seems to be more sensitive to them. This means that if compared to the 100 speaker model, they tend to be better, when trained using good parameters, but also worse, when trained using bad parameters.

MR	100 speakers	470 speakers
Good Parameters	5.19	**3.69**
All Parameters	**21.41**	23.94

Table 2. Averaged MR results of the angular margin loss models trained during the gridsearch in %.

3.2. Bottleneck Adaption

According to [4] and [6], a bottleneck layer using 512 hidden units has been used for their models. However, they had been trained on large scale datasets, consisting of over 10K classes and they explained that in theory, this dimension should be sufficient for far more classes. This puts our approach in question, as we have bottleneck dimensions of 500 for the 100 speaker set and 2'350 for the 470 speaker set.

We have tried to decrease said dimension as low as possible, while keeping up the performance of the model up to the state-of-the-art. In consideration of us being in a time rush, only models using a dimension of 3 in the bottleneck layer have been evaluated. Since the model learns a lot slower having such a small bottleneck, it had to be trained for 5'000 epochs to achieve acceptable performance.

The best model achieved a MR of 18.75% on the 40 speakers test set. Although this result is bad, we see room for improvement and anticipate results that keep up with the previous experiment.

3.3. VoxCeleb2

As we want to evaluate our model on real world data, we applied the proposed loss to the model introduced in [10]. It has the same structure as in the first experiment, but uses active learning rounds for training. It is not clear yet, what the optimal setting for this setup is, however, only a few models were been trained to get an idea of the performance of the proposed losses.

Loss Function	MR
PKLD	39.41
Angular Margin	**32.63**

Table 3. MR of the best models trained and evaluated on the VoxCeleb2 dataset in %.

We were able to improve the MR of the model by 6.78%, as visible in Table 3. It is important to know that we retrieved the MR differently for the PKLD loss. This leads to a value lower than the actual MR, which should be around 46% -

50%. Unfortunately, we were not able to retrieve the actual MR in given time, what we are very sorry for.

Nevertheless, we still wanted to give a more accurate impression of the influence of the proposed losses, since an improvement of around 16% is quite a change.

4. CONCLUSIONS

In this work, we introduced angular margin losses to speaker clustering applications for the first time and have evaluated its performance on the TIMIT and VoxCeleb2 dataset. We were able to prove that the losses improved state-of-the-art performance, thus are well suited for this research area as well. We see all the experiments as a success, but plan to further explore the second and third experiment, as the results clearly did not reach their full potential yet.

4.1. Future Work

We plan to decrease the complexity of the models while preserving the achieved performances of the first experiment, by conducting more research regarding the adaption of the bottleneck layer. We also see a possibility of completely removing the dense layer before the bottleneck. Therefore, we will be initiating another gridsearch to find the best configurations.

As mentioned, there is still a lot of work to be done for the models using the VoxCeleb2 dataset. We will conduct experiments on how to perform the training, as well as finding the best active learning setting, better utterance time and angular margin loss configurations.

Since the generation of a spectrogram transforms raw audio waveforms to a time-frequency domain, we simultaneously lose certain information about a speaker that may be better visible in the raw waveform. The WaveNet [12] is a generative model, that works on raw audio data, and has yielded great results in text-to-speech, multi-speaker speech generation, and speech recognition applications. As it seems to be possible to learn the characteristics of a speaker to generate speaker specific speech, we plan to train a WaveNet model on speaker recognition and speaker clustering. The WaveNet outperformed the state-of-the-art in speech recognition on the TIMIT dataset. Hence, we anticipate it to work for our problem setting too, as they are related. Altough the WaveNet being a model designed to process raw audio data, we still could feed it the spectrogram of the speech [13]. This could lead to interesting new knowlege, such as figuring out, if there is more information about a speaker encoded in the time-frequency domain, or in the raw audio waveform.

5. REFERENCES

[1] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren, "Timit acoustic phonetic continuous speech corpus cdrom," 1993.

[2] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *INTERSPEECH*, 2018.

[3] Jan Sonderegger, Patrick Walter, and Thilo Stadelmann, "Benchmarking of Classical and Deep Learning Speaker Clustering Approaches," Jun 2019.

[4] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou, "ArcFace: Additive Angular Margin Loss for Deep Face Recognition," *arXiv:1801.07698*, Jan 2018.

[5] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu, "CosFace: Large Margin Cosine Loss for Deep Face Recognition," *arXiv:1801.09414*, Jan 2018.

[6] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song, "SphereFace: Deep Hypersphere Embedding for Face Recognition," *arXiv:1704.08063*, Apr 2017.

[7] Yanick X. Lukic, Carlo Vogt, Oliver Drr, and Thilo Stadelmann, "Learning Embeddings for Speaker Clustering based on Voice Equality," Sept 2017.

[8] Thilo Stadelmann, Sebastian Glinski-Haefeli, Patrick Gerber, and Oliver Drr, "Capturing Suprasegmental Features of a Voice with RNNs for Improved Speaker Clustering," Aug 2018.

[9] Weidi Xie, Arsha Nagrani, Joon Son Chung, and Andrew Zisserman, "Utterance-level Aggregation for Speaker Recognition in the Wild," Apr 2019.

[10] Claude Lehmann, Christian Lauener, and Thilo Stadelmann, "Speaker Clustering for Real-World Data using Deep Learning," Jun 2019.

[11] Margarita Kotti, Vassiliki Moschou, and Constantine Kotropoulos, "Speaker Segmentation and Clustering," 2008.

[12] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu, "WaveNet: A Generative Model for Raw Audio," *arXiv:1609.03499*, Sep 2016.

[13] Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves, and Koray Kavukcuoglu, "Neural Machine Translation in Linear Time," *arXiv:1610.10099*, Oct 2016.