Guarding Against Deepfake Audio with One-Class Softmax

Chih-Yi Lin
Institute for Natural Language Processing (IMS), University of Stuttgart
st180953@stud.uni-stuttgart.de

1 Introduction

Current anti-spoofing detection methods face a generalization issue as they assume similar distributions for both bonafide and spoof classes between training and evaluation (test) datasets. However, this does not hold for spoofing attacks, as the techniques used to generate fake audios in the training data may never catch up with those used in the real world.

To tackle the generalization problem, Zhang et al. (2021), propose the One-Class Softmax (OC-Softmax) loss function, aiming to compact the real audios and push away the fake audios in embedding space through angular margin injection. They achieve the best EER performance among all single systems on the evaluation set of ASVspoof2019 LA subset.

In this project, we aim to explore the effectiveness of OC-Softmax on the deep fake audio detection task. Research questions include:

1. Can OC-Softmax outperform Softmax across diverse systems?

2. Does OC-Softmax exhibit superior generalization capabilities on real-world samples (In-The-Wild dataset) compared to Softmax?

2 Methodology

OC-Softmax works as follows:

$$L_{OCS} = \frac{1}{N}\sum_{i=1}^{N} log(1 + e^{\alpha(m_{y_i} - \hat{w_0}\hat{x_i})(-1)^{y_i}}) \quad (1)$$

where α is a scalar, m is the margin for cosine similarity, w_0 is the optimization direction of the target class embeddings, and $\hat{w_0}$, $\hat{x_i}$ are normalized w_0 and x_i. m_0 and m_1 denote the margin for the target class (bonafide) and non-target class (spoof), respectively, with $m_0, m_1 \in [-1, 1], m_0 > m_1$. That is, if the current data point x_i is from the bonafide class, m_0 forces the angle θ_i between x_i and w_0 to be smaller than $\arccos m_0$, while if x_i is from the spoof class, m_1 forces θ_i to be larger than $\arccos m_1$. OC-Softmax helps to compact the target class while pushing away the non-target class from the target class.

3 Datasets

Datasets	Descr.	#Train (Real/Fake)	#Dev (Real/Fake)	#Eval (Real/Fake)
ASVspoof 2019-LA	Artifacts from TTS or VC	25,380 (2,580/22,800)	24,844 (2,548/22,296)	–
ASVspoof 2021-DF	Compressed audios	100,000	25,000	–
In-The-Wild	Samples collected from the internet	–	–	31,779 (19,963/11,816)

Table 1: Summary of training, development and evaluation datasets.

The datasets used for training and evaluation include the ASVspoof2019 LA subset and the ASVspoof2021 DF subset for various models. For ASVspoof2021-DF, the bonafide class is oversampled (with replacement) to balance the two classes and then randomly sampled for the training and development sets, following the implementation from Kawa et al. (2023). The In-The-Wild dataset is exclusively used as the evaluation set for all models.

4 Features and Experimental Details

Feature	Model	Train/Dev Set	Window Len.(ms)	Hop Len.(ms)	# Filters	Time Frame	Embed Dim.
LFCC	ResNet18	ASV19-LA	20	10	20 dim * 3	7.5s	256
LFCC	ResNet18	ASV21-DF	400	160	128 dim * 3	15s	256
MFCC	ResNet18	ASV21-DF					256
MFCC + Whisper	MesoNet	ASV21-DF					1024

Table 2: The acoustic features used for training and experimental setup.

There are four experiment configurations for comparison, each utilizing different features including LFCC, MFCC, and MFCC+Whisper. Whisper, a pre-trained transformer-based encoder-decoder

ASR system, is employed for its encoder in the experiment as a feature extractor, concatenated with MFCC (Kawa et al., 2023).

In general, the features used for ASVspoof19-LA dataset are smaller in size compared to those used for ASVspoof21-DF, comprising LFCC or MFCC, delta, and double delta filters.

Regarding the experimental setup, all configurations are trained for 20 epochs, and the model with the lowest EER score on the development set is selected. For ResNet18, the learning rate is set to 0.0003 with a decay of 0.5 every 10 epochs, while for MesoNet, it is set to 0.0001 with a decay of 0.0001. Adam Optimizer is applied for all configurations. For the first configuration trained on ASV19-LA, the batch size is 64, whereas for the others, it is 8. The embedding dimension in the last column of Table 2 serves as the input for OC-Softmax. This dimension represents the intermediate representations before the last fully connected layer, and it varies depending on the model architecture.

5 Results

Feature	Model	Train/Dev Set	Loss	Dev	Eval
LFCC	ResNet18	ASV19-LA	Softmax	0.354%	**34.39%**
			OC-Softmax	0.279%	39.397%
LFCC	ResNet18 (100 epochs)	ASV19-LA	Softmax	0.274%	34.30%
			OC-Softmax	0.201%	**28.166%**
LFCC	ResNet18	ASV21-DF	Softmax	1.843%	60.176%
			OC-Softmax	2.196%	**47.157%**
MFCC	ResNet18	ASV21-DF	Softmax	1.464%	47.655%
			OC-Softmax	1.956%	**42.909%**
MFCC + Whisper	MesoNet	ASV21-DF	Softmax	0.444%	37.00%
			OC-Softmax	0.71%	**29.86%**

Table 3: EER is used as the evaluation metric for both the development and evaluation sets.

For all configurations trained for the same epochs, models using OC-Softmax outperform those using Softmax in all cases except for the first setting (LFCC-ResNet18-ASV19). However, after training the first setting for a longer duration, i.e., until 100 epochs, the model with Softmax did not show significant improvement, while with OC-Softmax, it continued to improve and eventually outperformed the one using Softmax. This suggests the potential performance gains of OC-Softmax with more training epochs across all configurations.

Models trained on the newer dataset, ASV21-DF, do not consistently outperform those trained on ASV19-LA, and they also exhibit higher Dev EER, indicating that these models may require more training epochs. This could be attributed to the larger training samples in ASV21-DF, as well as the use of larger and longer acoustic features compared to ASV19-LA. Nevertheless, with the help of Whisper-extracted features, MFCC+Whisper-MesoNet-OC-Softmax achieves the best performance for all configurations, suggesting the potential of using Whisper as a feature extractor.

Additionally, a significant EER disparity is observed between the development and evaluation sets, demonstrating what has been mentioned in the literature: the distributions between training and development sets are similar, while the distributions between training and evaluation sets are more dissimilar.

6 Visualization of Learned Embeddings

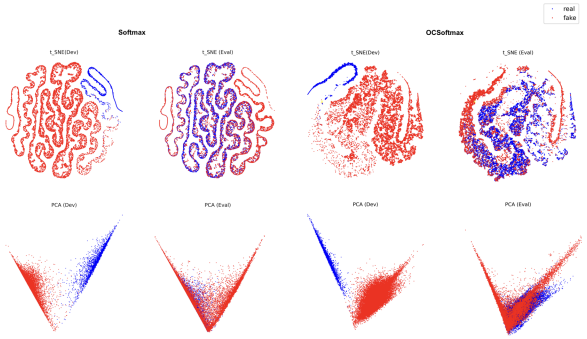

Figure 1: Distributions of the embeddings learned by LFCC-ResNet18-ASV19-100ep based on gold labels. The blue dots represent the data points from the bonafide class, whereas the red dots represent the data points from the spoof class.

The visualization in Figure 1 is created using t-SNE and PCA, employing the same coordinate system to compare the distributions of both classes with Softmax and OC-Softmax across the development and evaluation sets.

Firstly, distinct distributions are observed for both classes in the development and evaluation sets, as indicated by the differing positions of the blue and red clusters, e.g., in the two PCA plots at the lower left corner. This differs slightly from prior research, which only highlighted a distribution mismatch for the spoof class, without mentioning the mismatch of bonafide.

The second comparison focuses on Softmax versus OC-Softmax. While OC-Softmax results in a more compact cluster for the bonafide class in both development and evaluation sets compared to Softmax, it exhibits entanglement in the evaluation set. This presents a greater challenge for models,

consistent with the disparity observed between Dev EER and Eval EER in the previous section.

7 Conclusions

Based on the experiments outlined in this report, OC-Softmax outperforms Softmax across all settings on real-world samples, except for LFCC-ResNet-ASV19 trained for 20 epochs. However, after extending training to 100 epochs, this model with OC-Softmax also surpasses Softmax. Additionally, integrating Whisper's encoder as a feature extractor alongside traditional features demonstrates significant potential.

Future research directions include the following: 1. Evaluate models on additional real-world datasets to assess whether they yield similar performance. 2. Explore various *one-class classification* methods used in computer vision (Perera et al., 2021), such as employing GANs for fake image detection, which may also benefit deepfake audio detection. 3. Investigate *continuous learning* approaches, aiming to continuously learn from new datasets while retaining previously acquired knowledge, which could address the challenge of distribution mismatch between training and evaluation sets.

Acknowledgments

Thanks to Professor Thang Vu and Yixuan Xiao for supervising this research project.

References

Piotr Kawa, Marcin Plata, Michał Czuba, Piotr Szymański, and Piotr Syga. 2023. Improved deepfake detection using whisper features.

Pramuditha Perera, Poojan Oza, and Vishal M. Patel. 2021. One-class classification: A survey.

You Zhang, Fei Jiang, and Zhiyao Duan. 2021. One-class learning towards synthetic voice spoofing detection. *IEEE Signal Processing Letters*, 28:937–941.