CONTRASTIVE-CENTER LOSS IMPACT ON SPEAKER RECOGNITION

Project for deep learning course 046211 Technion institute of technology

Lior Bashari and Yonatan Kleerekoper

Index

1. Project overview

Speaker recognition is the process of identifying or verifying the identity of a speaker based on their voice. It is commonly used for security and authentication purposes, such as access control to secure buildings or computer systems. The main challenges in speaker recognition include variability in the speech signal due to factors such as speaking style, background noise, and microphone quality, as well as the need for large amounts of training data to accurately model an individual's speech patterns. Additionally, speaker recognition systems must be able to adapt to changes in a person's voice over time, such as due to aging or changes in health.

Our project's goal was to implement a speaker recognition neural network-based model that has similar accuracy to past work using similar methods [2], in addition, we wished to check the effect of contrastive center loss (CCL) on the performance of the model.

In our project, we implemented a ResNet-18 based neural network that trained on the VoxCeleb1 dataset. In order to adapt the audio files to fit the network we preformed the following augmentations:

1. An STFT was applied to the waveform.

2. The resulting spectrogram was then resized to fit the ResNet model expected size (224,224)
3. The spectrogram was the concatenated on itself to create a (3,224,224) Tensor.
4. The Tensor is fed into the ResNet-18 model with the CCL regularization.

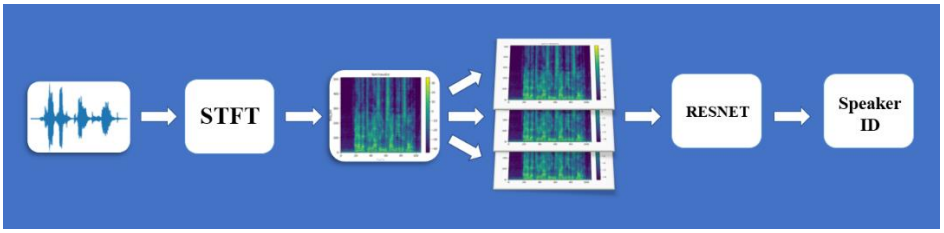


image 1: Block diagram of our model

In addition, in order to improve the models results we preformed additional preprocessing on the data and changed the ResNet-18 model, we will go over the changes and their effect in the Training and testing different augmentations section of this file.

2. Background

In this section we will go over the two main concepts of our project the first being the transformation to Mel-spectrograms and the second the Contrastive-center loss regularization.

2.1. Mel-spectrograms

A Mel-spectrogram is a way to represent audio data visually, with the x-axis representing time, the y-axis representing frequency, and color representing amplitude (loudness).

image 2: representation of the process from a 1D waveform to a 2D Mel-spectrogram.

It is similar to a traditional spectrogram, but it uses the Mel frequency scale instead of the linear frequency scale. The Mel scale is a non-linear scale that is based on the human perception of pitch, and it is often used in speech and music processing because it more closely matches our auditory system's response to different frequencies.

A Mel spectrogram is computed by Short-Time-Fourier-Transform,

$$STFT(t,w) = \sum_{n=-\infty}^{\infty} x[n]w[n-m]e^{-jwn}$$

$$MelSpectrogram(t,w) = \log\left(|STFT(t,w)|^2 + \varepsilon\right)$$

In our project, we used the librosa library's Mel spectrogram function with the following arguments:

- n_fft=512
- hop_length=160
- n_mels=40

Where n_fft is the window size of STFT calculation, hop_length is the hop size of STFT calculation and n_mels is the number of Mel scale frequencies to use.

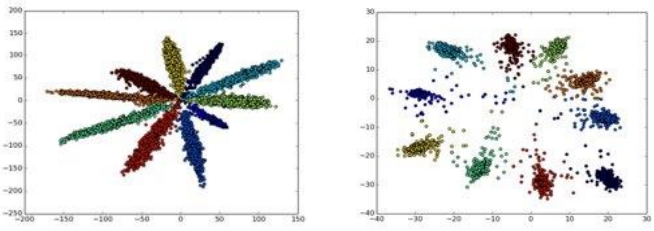
2.2. Contrastive-center loss regularization

Contrastive-center loss simultaneously considers intra-class compactness and inter-class separability by learning a center for each class. This loss penalizes the contrastive values between: (1) the distances of training samples to their corresponding class centers, and (2) the sum of the distances of training samples to their non-corresponding class centers.

$$\mathcal{L}_{CCL} = \frac{1}{2}\sum_{i=1}^{m} \frac{\left\|x_i - c_{y_i}\right\|_2^2}{\left(\sum_{j=1, j\neq y_i}^{k}\left\|x_i - c_j\right\|_2^2\right) + \delta}$$

Where m denotes the number of training samples in a minibatch. $x_i \in R_d$ denotes the i_{th} training sample with dimension of d. d is the feature dimension. y_i denotes the label of x_i. $c_{y_i} \in R_d$ denotes the $y_{i_{th}}$ class center of deep features with dimension d. k denotes the number of classes.

Ce et al [3] also demonstrate the effect of using this regularization on the MNIST dataset:



Cross entropy loss Contrastive Center loss

image 3 The effect of CCL on the MNIST dataset [3].

3. The Dataset

In our project we used the VoxCeleb1 dataset. VoxCeleb1 is a dataset of speech snippets used for training and evaluating speaker recognition systems. It was created by researchers at the University of Oxford and consists of over 100,000 clips of audio from over 1,251 celebrities. The dataset includes a diverse set of speakers with different accents, ages, and genders, and includes both studio and telephone-quality recordings. The dataset is widely used in the research.

Due to computational and memory limitations, our model was trained on the first 200 speakers (ids 1 to 200) but will work on a larger section of the dataset. The final train-validation-test split was:

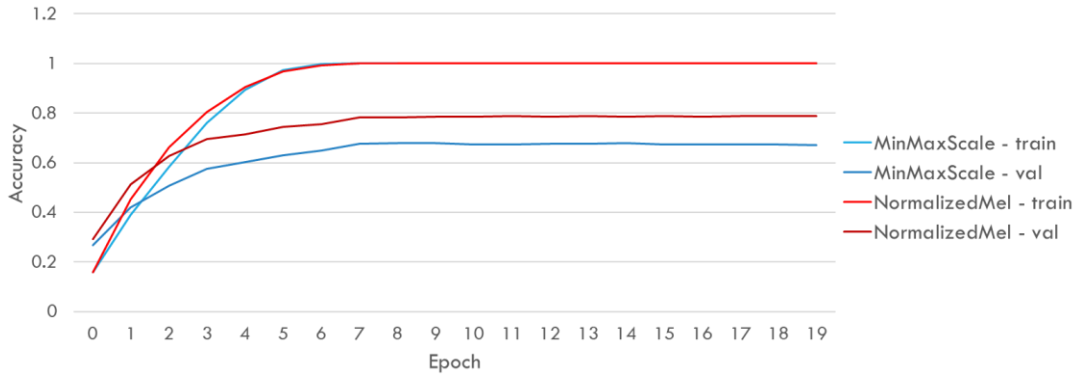
	Train	Validation	Test
#Audio files	14285	4761	4763

Table 1: train-validation-test split of the audio files in a 60%-20%-20% split.

4. Results and analysis

4.1. Data augmentations

Firstly, in order to create a good baseline for our project, we wanted to create a model with similar accuracy to the models seen in previous work [2]. In order to obtain this, we tested two different normalization methods as data augmentations. The first being min-max scaling and the second normalizing by the frequency bins. It is important to note, that normalization is critical for the model to work properly since Resnet-18 expects a normalized input. We obtained the following



graph 1: The train and validation scores for the different normalizations.

As seen above, both normalizations get an 100% accuracy on the train set, but have a large deficiency in the validation set. The frequency bins normalization got an accuracy score that is 15% higher than the min-max scaling normalization.

Therefore, we chose our baseline to be the network where we normalize the input according to the frequency bins. This model has similar accuracy to the models shown in Jakubec et al's paper [2].

4.2. Examining the impact of Contrastive-center loss regularization

4.2.1. Hyper-parameters tuning

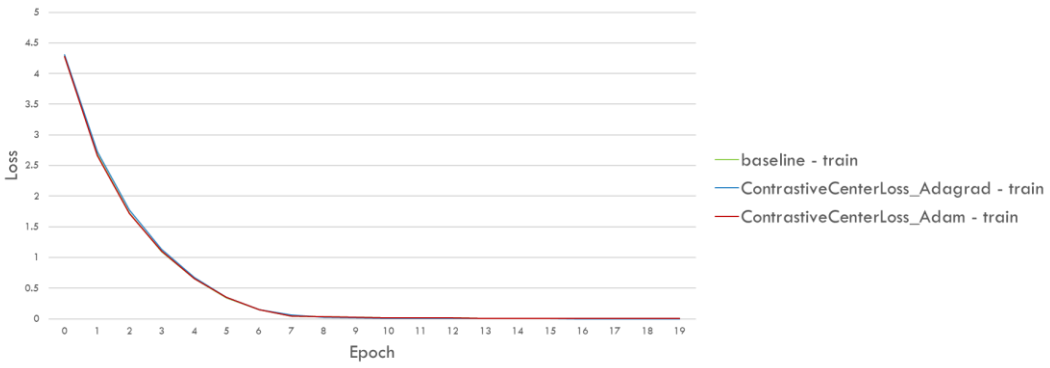
Next, in order to find the best parameters to run with the CCL regularization, we preformed hyperparameter tuning using Optuna library, where the hyperparameters we checked were the optimizer, learning rate of the CCL optimizer and the λ regularization coefficient. We then ran two experiments the first with the parameters from the Optuna run, and the second, with the coefficients from the Ce et al's paper [3].

	λ	Optimizer	Learning rate
Ce et al	1.0	Adagrad	0.001*
Optuna	0.55	Adam	0.002

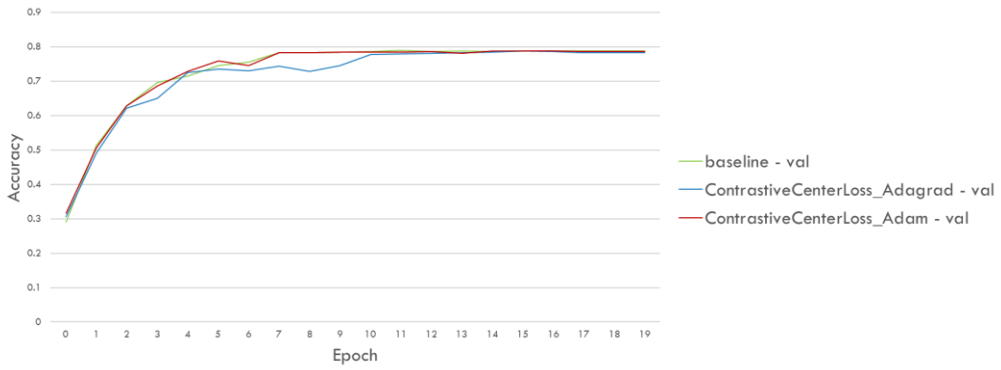
Table 2: The values we used in our experiments
** Ce et al did not publish their learning rate, we chose one that gave us convergence.*

4.2.2. Training models

We trained these experiments and got the following results,



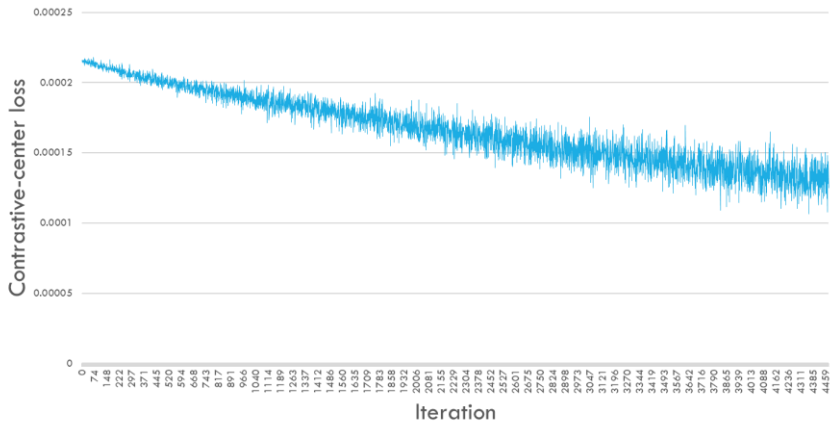
graph 2: The loss on the trainset as a function of Epoch for our three different experiments



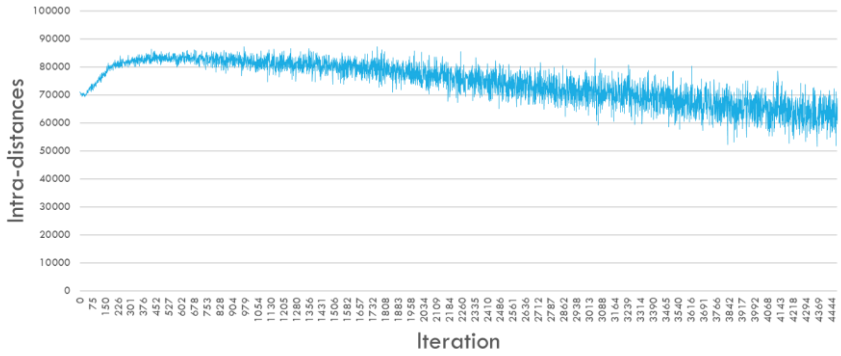
graph 3: The accuracy on the validation set as a function of Epoch for our three different experiments.

From the results above we see that the CCL regularization does not affect the train set converge. In addition, in all cases on our validation set, we converge relatively to the same value but in a different pace.

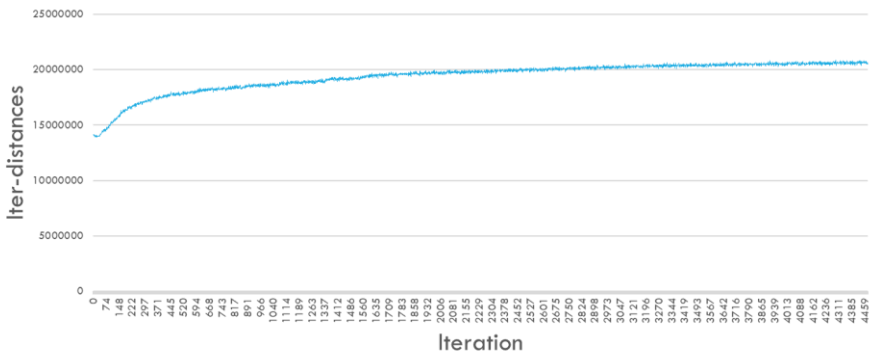
To understand the CCL regularization impact, we delve deeper into the CCL components.



graph 4 Contrastive-center loss as a function of iterations



graph 5 The values of the CCL loss intra-distances, $\frac{1}{2}\sum_{i=1}^{m}\left\|x_i - c_{y_i}\right\|_2^2$, as a function of iterations



graph 6 The values of the CCL regularization inter-distances, $\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1,j\neq y_i}^{k}\|x_i - c_j\|_2^2$, as a function of iterations

As we can see from graph 4, the value of the loss is low from the first iteration - approximal $2 \cdot 10^{-4}$ as opposed to the cross-entropy loss that starts at the value of about 4. **Due to the extreme size difference between the CCL regularization and the Cross Entropy loss, the CCL regularization has a minor impact on the train and validation coverage as seen in graphs 2 and 3.**

In addition, the low value of the CCL suggests that our data has a good ratio between class compactness and classes separability from the beginning of the learning process. We also see that even though the affect is little, we do increase the compactness and separability of the data by learning better centers of the classes.

4.2.3. Evaluating models

Our models have been evaluated by top-1 and top-5 accuracies on the test set,

Model	Top-1 (%)	Top-5 (%)
Baseline	**79.09**	92.78
Contrastive-center loss with Adagrad	78.98	**93.24**
Contrastive-center loss with Adam	78.52	92.82

Table 3 Models' scores on test set

As seen in the table above, all 3 experiments obtain similar top-1 accuracy, whereas **the Contrastive-center loss experiment with Adagrad optimizer increased the top-5 accuracy by 0.5%, showing that the regularization improved the generalization of the model**.

5. Conclusion and Future work

As we have shown this regularization function does have potential in improving models for these kinds of tasks, in order to further research this subject and improve the accuracy of the model, we propose the following steps:

1. Run a larger experiment using the full dataset. It is important to note that the full dataset is balanced between men and women and the split we used was not necessarily balanced. This might have led to decrease in performance. In addition, we feel that the larger the dataset the less separable it is, meaning the regularization will have a larger affect.

2. Gain better use of the dataset by clipping and splitting all the files to the same size. This will grow the dataset and will lose less data when resizing the Mel-spectrogram in order to fit into the model, potentially increasing the accuracy.

3. The Contrastive-center loss represent the ratio between the compactness of each class and the separability between classes. As seen in the Results section, the ratio is very small, but each component by itself is rather large. We propose to divide the Contrastive-center loss into a sum of its two components: intra-distances and inter-distances. This way if both of the distances are rather large they will have a greater affect on the entire loss function. Note that these distances are of a large dimension, 512, therefore, it is needed to take into account.

6. References

We based our project on the results of the following papers and github repositories:

[1] S. Bianco, E. Cereda and P. Napoletano, "Discriminative Deep Audio Feature Embedding for Speaker Recognition in the Wild," 2018 IEEE 8th International Conference on Consumer Electronics - Berlin (ICCE-Berlin), Berlin, Germany, 2018, pp. 1-5, doi: 10.1109/ICCE-Berlin.2018.8576237.

[2] M. Jakubec, E. Lieskovska and R. Jarina, "Speaker Recognition with ResNet and VGG Networks," 2021 31st International Conference Radioelektronika (RADIOELEKTRONIKA), Brno, Czech Republic, 2021, pp. 1-5, doi: 10.1109/RADIOELEKTRONIKA52220.2021.9420202.

[3] Qi, Ce, and Fei Su. "Contrastive-center loss for deep neural networks." 2017 IEEE international conference on image processing (ICIP). IEEE, 2017.

[4] https://github.com/samtwl/Deep-Learning-Contrastive-Center-Loss-Transfer-Learning-Food-Classification-/tree/master