Assignment: Homework 6

What to Hand In

1. Codes that can reproduce all your results. It is your responsibility to make all your codes readable. Detailed comments and meaningful variable names are highly recommended.
2. Your saved trained model and the script to run your model.
3. A report (PDF) that describes your attempts. You should include what you have done and discussions on cases when your model works and fails, along with possible explanations and potential improvements. You are free to cite published papers in your report, but it is prohibited to cite their code in your code.
4. Any other files if necessary.

How to Hand It In

1. Put all your solutions, including a report showing your answers and figures, in one folder. Compress this folder and name it <firstname>_<lastname>_HW6.zip. For example, “Zhiyao_Duan_HW6.zip”.
2. Submit to the appropriate entry on Blackboard.

When to Hand It In

It is due at 11:59 PM on the date specified on the course calendar. Late assignments will receive a 20% deduction of the full grade each day.

Problems (10 points in total)

This homework will guide you through the process of applying neural networks to two speech processing tasks: speaker verification and speaker diarization. The basic approach is as follows:

1. Train a speaker identification network, classifying speech signals according to the speaker labels. Use the output before the last layer as speaker embeddings.
2. Speaker verification: Compare the speaker embedding of paired speech signals and compute the cosine similarity.
3. Speaker diarization: Perform clustering on the extracted speaker embedding from the speech segments based on the speaker embedding similarity.

A Jupyter notebook containing foundational skeleton code is provided to kickstart your project. While this code serves as a basic guideline, you are encouraged to explore beyond its structure. The notebook already includes a set of necessary packages for this assignment. However, should you find the need to incorporate additional packages to augment your work, please submit a request for approval on Piazza by making a public post.

If you get stuck with the training, feel free to skip questions 1 and 2b), for the rest of the homework, you will only need to use a pre-trained speaker embedding extraction network.

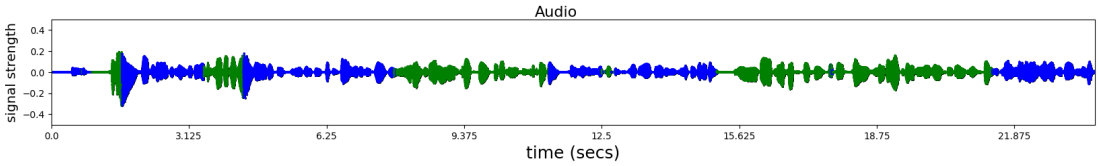
1. (3 points) Speaker embedding extraction. Train a speaker identification network with the CREMA-D dataset.

a) (1 point) Dataset. Construct the Dataset class for the CREMA-D dataset. You only need to download the “AudioWAV” folder from https://github.com/CheyneyComputerScience/CREMA-D. Split the dataset into training and validation sets with a ratio of 9:1. Construct the Dataloader for them respectively.

b) (0.5 points) Construct the classification model using ECAPA-TDNN. ECAPA-TDNN is a speaker embedding extraction network. You will need a linear layer appended to it.
c) (1 point) Training and validation loop. Train the model with the classification objective (cross-entropy loss) and observe the classification accuracy on the validation set.
d) (0.5 points) Save the best-performing model on the validation set for use as the speaker embedding network.

2. (2.5 points) Speaker Verification.
a) (1.5 points) Construct a list of paired data with a length of 1000 from the validation set of the CREMA-D dataset with labels for speaker verification. If they are from the same speaker, then the label is 1; otherwise, it is 0. Calculate EER on the list you constructed.
b) (0.5 points) Load the pre-trained model from https://github.com/TaoRuijie/ECAPA-TDNN and report the EER on the list you constructed in 2a).
c) (0.5 points) Discussion. Analyze the performance differences between your model and the pre-trained model, considering the training data used. Discuss the potential reasons for their performance difference.

3. (4.5 points) Speaker Diarization. For this question, you can directly use the pretrained model. If you are interested, you can also apply your trained model in question 1 here and compare the performance.
a) (0.5 points) Load the provided conversation audio file. Segment the audio into smaller, manageable parts suitable for processing. For each segmented audio part, calculate the speaker embeddings.
b) (1 point) Cluster the embeddings with k-means. Visualize the results with a colored waveform where different colors represent different speakers (An example is shown below), and listen to the sample again. Are you satisfied with the results?



c) (1.5 points) Evaluation. Construct an RTTM file with your prediction. Calculate the Diarization Error Rate (DER) using the official DER calculation tool from https://github.com/ftshijt/dscore. Implement DER manually with the formula. Compare with the results from the official tool. Answer the question: Is the upper limit of DER 100%?
d) (1.5 points) Further thinking and implementation. Do you see any limitation of the methods we are using for the diarization on more general conversations? (Hint: more number of speakers, more overlapped speech, more silence in between, more background noise, etc.) Explain the disadvantages and how you would like to improve upon them. Implement some of your ideas and discuss the results. If it is too complicated to implement and requires additional data for training, describe the steps in detail.