EE798K Assignment 4: Speach Detection Model

Chinmay Hiran Pillai
200298
(Dated: September 30, 2023)

The paper explores a speech recognition model that utilises Mel-Frequency Cepstral Coefficients (MFCCs) and Deep Neural Networks (DNNs) to detect speech from audio signals. It first outlines the architecture and features of the proposed model, then delves into the critical requirements necessary for real-time processing. The discussion then extends to potential challenges faced during implementation and provides strategies to mitigate these issues.

I. INTRODUCTION

Mel-Frequency Cepstral Coefficients (MFCCs) have played a vital role towards in speech recognition as they are crucial features that efficiently represent the power spectrum of audio signals, serving as a pivotal component in the development of effective ASR systems [2]. Utilisation of Deep Neural Networks (DNNs) have also contributed heavily towards their advancement due to their ability to learn complex features really well [1].

MFCCs serve as a reliable feature extractor, capturing the essential phenotic characteristics of the acoustic signal, while DNNs offer robust pattern recognition capabilities, allowing the system to understand and interpret these features accurately.

II. METHOD DESCRIPTION

The model involves first splitting the audio into short-time frames based on window size (w) and hop size (h). We then need to classify each frame as containing speach or not, i.e run binary classification over them. We utilise Deep Neural Networks (DNN) to classify the frames because of DNNs' ability to handle complex mappings between acoustic signals and phonetic classifications more effectively, thereby enhancing the performance of ASR systems [1].

For the input for the classification, we need to provide the network with features that capture that phonemes unique to speech which can be utilised to classify them. For this we use MFCCs since they capture the features specific to speech and phonemes.

For even better performance, we will also provide the first and second differences of the MFCCs (ΔMFCCs and Δ^2MFCCs) in order to capture the features corresponding to how the MFCCs and hence the phenomes are changing thoughout the audio. This should give the DNN more relevant information to compute more accurate results. Since the number of MFCCs used are genrally small, calculating the their first and second differences won't be relatively computationally expensive either.

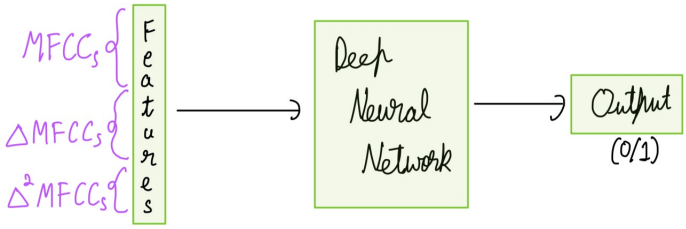

FIG. 1. Model for classifying each frame

Once every frame has been classified as containing speech or not (1/0), we can find the start and end times of speech using the following code,

```
#If previous frame has speech
if previous_frame == 1:

        #and current frame doesn't have speech
        if current_frame == 0:
            print("Start Time: ", start_time,
                  "End Time: ", current_start_time)

#If previous frame doesn't have speech
#and current frame does have speech
elif current_frame == 1:
        start_time = current_start_time
```

FIG. 2. Code to run on every frame for finding start and end times

where, "current start time" is the starting time for the current frame and "current frame" and "previous frame" are the values outputted by the DNN (0/1) for the current and previous frame respectively.

III. DISCUSSION

A. Real-Time Computation

For real-time computation we need to ensure that the execution time for processing the audio data till a particular time is less than or equal to the particular time, i.e (number of windows processed at any current time * Time required for processing 1 window <= Current time)

$$(floor(\frac{t-w}{h})+1)*T <= t \tag{1}$$

where, T = Time required for processing 1 window
t = current time
w = window size
h = hop size

If the condition without the floor is satisfied, the above condition is also satisfied since the floor of a number is less than or equal to the number. Then the condition for real-time computation become,

$$(\frac{t-w}{h}+1)*T <= t \tag{2}$$

$$\implies \frac{t*T}{h}+(1-\frac{w}{h})*T-t <= 0 \tag{3}$$

$$\implies t*(\frac{T}{h}-1) <= T*(\frac{w}{h}-1) \tag{4}$$

Since we maintain window size to be greater than the hop size, making $\frac{w}{h} > 1$, we can observe that the RHS of the equation is positive. Since $t > 0$, the condition for satisfying the equation for any $\frac{w}{h}$ ratio is making $(\frac{T}{h}-1) <= 0$. Only then will the condition satisfy $\forall w > h$.

Hence, for real-time computation, we require,

$$\frac{T}{h} <= 1 \tag{5}$$

$$\implies \mathbf{T <= h} \tag{6}$$

Therefore, for real-time computation, we require the time required by the algorithm to process 1 window to be less than the hop size of the spectrum. We hence need to choose our DNNs architecture and hop size (and hence window size) carefully in order to enable real-time processing.

B. Gaps in Speech

Pauses are a common occurance during speech and we need our model to be capable of recognising them as parts of the ongoing speech instead of classifying them as a break in speech. We can achieve this by optimizing our window size to be bigger than the duration of a typical pause in speech but keeping it small enough to achieve good precision.

C. Hyperparameters

The hyperparameters for the system are:

1. Hop Size (h): Optimize for real-time computing as mention above
2. Window Size (w): Optimize for presion, accuracy and computation time as mentioned above
3. Architecture of the DNN: Optimise for accuracy while maintaining enough speed for real-time processing
4. Number of MFCCs: Optimise for accuracy
5. Learning Rate of DNN: Optimise for convergence to global minima of DNN's loss function

IV. APPLICATIONS

There are many application for Speech Detection, especially one which is fast enough for real-time processing. Some of the apllications are [3]:

1. **Virtual Assistants**: Virtual assistants use ASR to convert user voice commands into text, enabling user-device interaction through natural language.
2. **Accessibility Tools**: ASR can be crucial for developing tools and technologies to assist individuals with disabilities.
3. **In-Vehicle Voice Controlled Systems**: In-vehicle systems employ ASR to allow drivers to control navigation, music, and calls using voice commands, promoting hands-free operation.

V. CONCLUSION

The paper explored a model for speech detection using MFCCs and DNNs. We then delved into the model's requirments for real-time processing, potential problems in implimentation, how to mitigate them and finally the applications of such a model.

[1] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97, 2012.

[2] Xuedong Huang, Alex Acero, Hsiao-Wuen Hon, and Raj Reddy. *Spoken language processing: A guide to theory, algorithm, and system development*. Prentice hall PTR, 2001.

[3] Biing-Hwang Juang and Lawrence R Rabiner. Automatic speech recognition–a brief history of the technology development. *Georgia Institute of Technology. Atlanta Rutgers University and the University of California. Santa Barbara*, 1:67, 2005.