Deep learning-based detection of Dysarthric speech disability

Stanford CS224N Custom Project shared with course CS230

Siddhartha Prakash
Department of Computer Science
Stanford University
sidp@stanford.edu

Abstract

Dysarthria is a form of speech disability resulting from neurological damage to the motor component of motor-speech system of the brain. It results in lack of control over the muscles generating speech and the speaker has slurred, incoherent, poor articulation of phonemes and hard to decipher speech characteristics. The recorded sound is stored in digital format with two dimensions of frequency and time. There are multiple ways in which this sound signal can be processed like converting it into formats like two-dimensional image, spectrogram, filter banks, Fourier Transform. We combine three things – differences in speech pattern of dysarthric and non-dysarthric persons, different ways of representation of this speech signal and the use of deep learning to learn the pattern. The end goal of the project is to differentiate dysarthric and non-dysarthric audio signals. To achieve this, we use some additional processing of the input data, feed it into a Convolutional Neural Network based model, and train the model output through a softmax classifier. Training and testing it on TORGO database [1] containing dysarthric and non-dysarthric speakers with text data containing their speech content, we were able to get the best result of detecting dysarthric at 68% accuracy on test set of the audio signal converted to mel-spectogram. The result was different based on audio processing method used as well as on whether we measure single word audio input or sentence level audio input. We demonstrated the importance of speech encoding and leave open future effort in this area to improve results of deep learning models on speech processing.

1. Introduction

Most modern Automated Speech Recognition (ASR) technologies as well as communication technologies ignore the situation of people impacted by Dysarthria, which causes slurred and incomprehensible speech. A person suffering from it cannot be an active participant of a conference call because no one else will understand what the person is speaking. Dysarthria affects 170 per 100, 000 persons and almost one third of persons with traumatic brain injury [2].

Since the speech of different people with different accent, style of speaking, varying levels of speech disability are different, detecting dysarthria disability is a challenge. Manually detecting dysarthria is costly, error prone, time-consuming and subjective. A small part of the speech signal of a dysarthric person might sound similar to normal speech of another person without dysarthria with a different accent. Representation of speech for processing, unlike text, is not yet matured to easily allow such differentiation particularly when using it to train deep learning models.

Current methods of detecting dysarthria have primarily focused on designing a optimal model and discovering its hyperparameters to get best results. Most of the research efforts have not taken into account the impact of processing and encoding of the speech signal so that the models can learn the features more easily. This motivated us to look into various methods of speech encoding, study its effect on models to detect Dysarthria. We were able to establish that for our given dataset, mel-spectrogram conversion of the sound signal before feeding into a CNN based deep learning model was the best representation of the audio signal for detecting Dysarthria. For input processing we used three methods – separate sentence level and word level audio signals, perform z-score normalization and padding and finally use four known audio signal representation namely STFT (Short Time Fourier Transform), Mel-Filterbank, Spectrogram, Mel-Spectrogram.

2. Related Work

[3] used simple linear classifiers trained on TORGO database to detect Dysarthria. Efforts to detect the disability include [4] where the input signal was converted into time domain filterbanks and passed through a set of LSTM and fully connected layers in an attention based model and the hyperparameters tuned to get a success rate of 72% on TORGO database. [2] tried to regenerate normal speech from dysarthric by passing them through a Cycle consistent GAN adversarial training model by converting the speech into spectrogram image of 128x128 with padding. They were able to gain about 30% higher intelligibility of reconstructed speech compared to raw speech. Using deep learning model with spectral representation of speech as an input has been attempted by [5], [6], [7] and [8] while [9] did it and also applied mean variance normalization to the input audio signal. Another good attempt to detect Dysarthria and reconstruct the dysarthric speech into intelligible speech was done by Daniel Korzekwa et al in [11]. The model is fed input in form of mel-spectrogram of the audio signal while both detection of dysarthria and reconstruction of normal speech from dysarthric speech are trained together. The detection of dysarthria is done using a CNN based segment while reconstruction of original Mel-Spectrogram is done by combining processed audio-signal and text input together into a latent space called ‘Dysarthric Space’. This allows the reconstruction part of the model to use text as well as audio signal to learn the features required for reconstruction. [12] showed that for raw time series data, the CNN layers have to be very deep, up to 24 levels, for the deep learning model to learn and differentiate audio signals. And the same can be achieved with fewer CNN layers if we use spectrograms instead of raw waveforms as shown by [13]

In almost all the efforts described there are three basic building blocks: process the input signal, feed the signal into a deep learning model, learn the parameters of the model for dysarthria detection.

3. Approach

3.1 Model architecture

The model used is a variation of the model used by [11] and three notable changes. First, we focused on the processing of the input data. The original model only uses mel-filterbank representation of the acoustic signal. We changed it to try the four standard representations of the acoustic signal namely, Short Time Fourier Transform (STFT), Mel-Filterbank, Spectrogram and

Mel-Spectrogram. Second, we replace the GRU layer in the model with LSTM layer. Third, we break out the Dysarthria detection part of the original model from the speech reconstruction part and train the detection part only.

The audio signal is converted into its representation followed by z-score normalization. We pad the audio signals to have consistent 5 second audio length. Audio signals longer than 5 seconds are trimmed at 5 second length so that we have a consistent length of the audio. This signal representation dataset is then fed in batches into two layers of two-dimensional Convolutional Neural Network with 20 channels, 5x5 kernel and RELU. The CNN layers inherently have two-dimensional Batch Normalization as well as two-dimensional MaxPool layers in them. The output from this layer is fed into an LSTM layer. Note that the original model in [1] had a GRU layer for simplicity, but we decided to keep LSTM layer because since we have removed the reconstruction part of the model, even with LSTM the model is simple enough for our training. The output from LSTM layer is passed through two levels of dense bottleneck layers followed by a Softmax layer. We applied a drop off value of 0.3 on CNN layers, LSTM layer and dense bottleneck layers. The loss was calculated using BCELossWithLogits in Pytorch which combines Sigmoid layer and Binary Cross Entropy Loss in one single class. The cross-entropy loss is calculated using

$$\sum_{i=1}^{m} \alpha log(p(d_i|X_i, \theta)) .$$

where the logarithmic part is the cross entropy between predicted and actual labels of the data. The model is initialized with Xavier method and trained using min-batch Stochastic Gradient Descent with a batch size of 4, learning rate of 0.001 and momentum of 0.09. Hyper-parameters of the model were tuned using manual tries with different sets of variations and the result reported is only with the best score we found of the set that we tried.

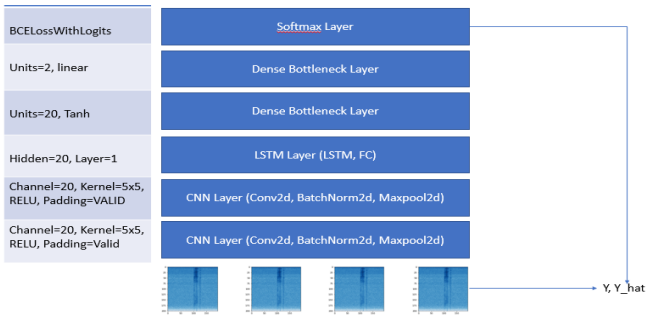

Figure 1: Model diagram representation with audio representation as input

The implementation of model itself was done entirely from scratch using Pytorch because the authors of [11] have not made their code public. Audio reading and conversion into datasets to feed into model was done using Librosa library and TorchAudio library. Audio processing part of implementation uses some learning from Audio Classifier Tutorial of Pytorch while conversion of audio from its waveform to its various two dimensional representations used library functions implemented in [10].

3.2 Baseline

We use baseline data from [4] because we have a common input data source which is TORGO database. The score they generated was averaged over 3 runs and had a best score of 72.4% with 3% of variability with mel-spectrogram as input. One key difference with our approach was that they divided the input data based on folder names. Since the folders represent the session of recording of the data, thus folder "FC01" represents one session of recording by one single person and hence is likely to be consistent in character potentially causing an artificially inflated score, it doesn't have good randomization. On the other hand, for our experiment, we took the entire dataset of about 8000 audio samples containing both dysarthric and non-dysarthric speakers which are both male and female and shuffled them before generating our training and test dataset.

4. Experiments

4.1 Data

We use the data from TORGO database [3] which is about 18 GB of data containing audio, video, text and others which was generated and published by University of Toronto. The data contains folders for each speaker, there are male and female speakers with both dysarthric and non-dysarthric voice samples. For each audio file, there are corresponding text files having the text prompt containing the words the speaker is speaking. The input data is preprocessed in three stages. In first stage, there are some audio files which are corrupt and hence couldn't be loaded using either librosa or torchaudio library, so those files and corresponding labels are removed from the dataset. In second stage, the dataset is divided into those with single word in the audio signal and those which are sentences. This classification is done based on the number of words in the text label corresponding to the audio file. There are about 6500 single word records and about 2500 multiple word records. The third level of processing is to read the audio, perform z-score normalization and padding or trimming based on the length of the audio file. If the length of the audio file is less than 5 seconds, we perform padding to make it 5 seconds, else we perform trimming.

Label generation is done based on the folder names. The data is grouped together based on dysarthric and non-dysarthric speakers. So non-dysarthric speakers are present in folders 'FC' and 'MC' where F stands for female speaker, M stands for male speaker, C stands for control group. Correspondingly, dysarthric speakers are present in folders 'M' and 'C'. Based on the folder nomenclature, one of data contains the text label, audio file path and label of 1 or 0 determining whether it is dysarthric or not.

4.2 Evaluation Method

Being a classification problem, we first tag all the samples as dysarthric or non-dysarthric. Let (X, y) represent one sample of the training set. X here represents the audio from speaker which can be in any encoded format like Mel-Spectogram in which case it will be a 2D space. $y \in \{0, 1\}$ represents label as non-dysarthric or dysarthric. For all set of training and test runs, we pick the score after a few iterations of runs and manually training the hyperparameters by trial and error.

The score is the percentage of total test set data, where the model is able to accurately predict the label.

4.3 Experimental Details

We generate a total of 12 set of results. The input to the model is passed in four formats - Short Time Fourier Transform (STFT), Mel-Filter bank, Spectrogram, Mel-Spectrogram. The four representations are each used on the three groups of data – single word audio, multiple word audio, combined set. The overall data is divided into training, validation and test set in the ratio of 95, 2.5 and 2.5 respectively over an overall dataset of about 9000 audio files for the combined case of word and sentence category data mixed together. Note again that the baseline data we have, though not exactly in similar dataset, is using Mel-Spectrogram and combined set. Overall training time in general for one set of experiment with one set of pre-defined hyperparameters was 2 hours.

4.4 Results

The result of experiments is tabulated in table 1, rounded to nearest full unit value.

	Words only	Sentence only	Word and sentence Mixed	Reference
Mel-Spectrogram	68%	62%	66%	72 +- 3% [4]
STFT	58%	54%	56%	NA
Spectrogram	63%	58%	61%	NA
Mel-Filterbank	58%	48%	52%	NA

Table1: Result of running the model with different audio processing.

5. Analysis

Based on the results, we observed that depending the audio processing technique applied, the results can vary. The hyper-parameter tuning requirement is also different based the different type of input processing. While we were not able to get the best reported result so far, we did notice that the results are best for Mel-spectrogram conversion of the audio compared to any other conversion. The hyper-parameter tuning could be one reason for this and the idea that another set of hyper-parameters could result in better score for say Spectrogram cannot be ruled out. Another key observation is that for words only, the detection is generally higher than for sentences. This could be either because the word is a simpler representation of the audio signal and hence classification learning is easier for the model. Another reason could be the padding and cut-off that we did to keep all files at 5 seconds. Even listening manually, the slurring for multiple word sentences was different compared to single word sentences.

6. Conclusion and future work

We were able to establish that the audio signal encoding method used as input to a deep learning model during training affects its performance. We also established that Mel-Spectrogram is generally the best choice for classification for the task of detecting Dysarthria, though it might be different for training of other models. For future work, we have three suggestions – First is to use

the processing of the audio as a layer in the deep learning model so that depending on the type of model being training, the training process is able to pick up the best audio processing method to use. Second is to try to create richer set of Audio encoding than the four most popular options listed above, one option can be to try to represent audio in three or more-dimensional space instead of the two-dimensional space that all these four methods have. Third is to try to regenerate normal audio signal from dysarthric signal after detection of dysarthria in the audio. This regenerated audio can then be compared with control set audio and difference will be likely easier to catch.

7. References

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in neural information processing systems, 2012, pp. 1097–1105.

[2] Seung Hee Yang and Minhwa Chung, "Improving Dysarthric Speech Intelligibility Using Cycle-consistent Adversarial Training" 2020.

[3] Jangwon Kim, Naveen Kumar, Andreas Tsiartas, Ming Li, and Shrikanth S. Narayanan, "Automatic intelligibility classification of sentence-level pathological speech," Computer Speech & Language, vol. 29, no. 1, pp. 132 – 144, 2015.

[4] Millet, Juliette; Zeghidour, Neil, "Learning to Detect Dysarthria from Raw Speech", in International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019

[5] Yedid Hoshen, Ron J Weiss, and Kevin W Wilson, "Speech acoustic modeling from raw multichannel waveforms," in Proceedings of ICASSP. IEEE, 2015.

[6] Tara N Sainath, Ron J Weiss, Andrew Senior, Kevin W Wilson, and Oriol Vinyals, "Learning the speech front-end with raw waveform cldnns," in Interspeech, 2015.

[7] Pegah Ghahremani, Vimal Manohar, Daniel Povey, and Sanjeev Khudanpur, "Acoustic modelling from the signal domain using cnns," in INTERSPEECH, 2016

[8] Neil Zeghidour, Nicolas Usunier, Iasonas Kokkinos, Thomas Schatz, Gabriel Synnaeve, and Emmanuel Dupoux, "Learning filterbanks from raw speech for phone recognition," 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5509–5513, 2018.

[9] Neil Zeghidour, Nicolas Usunier, Gabriel Synnaeve, Ronan Collobert, and Emmanuel Dupoux, "End-to-end speech recognition from the raw waveform," in Interspeech, 2018

[10] https://github.com/keunwoochoi/torchaudio-contrib

[11] Daniel Korzekwa, Roberto Barra-Chicote, Bozena Kostek, Thomas Drugman, Mateusz Lajszczak, "Interpretable Deep Learning Model for the Detection and Reconstruction of Dysarthric Speech" in 20th Annual Conference of the International Speech Communication Association, 2019, arXiv:1907.04743v1 (2019)

[12] Wei Da, Chia Dai, Shuhui Qu, Juncheng Li, Samarjit Das "Very deep convolutional neural network for raw waveforms.", in International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019

[13] Karol J Piczak, "Environmental sound classification with convolutional neural networks," in 2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP). IEEE, 2015, pp. 1–6.