Neural Network Project: Emotion Detection from Speech

ECE 5268 Theory and Application of Neural Networks

Alesandra Wiechecki Vergara

Florida Institute of Technology

April 29, 2024

Introduction

Outline

- Problem of Interest: Emotion Detection from Speech
- Data Resource
- Original Analytic Approach
  1. Waveform Analysis Package
  2. Mel Frequency Cepstral Coefficients (MFCCs)
- Baseline Model
  1. Architecture
  2. Performance
- Proposed Model
  1. Architecture
  2. Performance
- Project Summary
- References

Problem of Interest: Emotion Detection from Speech

Sentiment Analysis is the process of analyzing digital text to determine if the emotional tone of the message is positive, negative, or neutral.

More specifically, sentiment analysis is the use of *natural language processing*, *text analysis*, *computational linguistics*, and *biometrics* to systematically identify, extract, quantify, and study effective states and subjective information.

Examples:

- **Customer Support**
  - Better Address Customer's Needs
- **Crisis Call Center**
  - Real-time Response of Caller's Emotional State

For more information regarding Sentiment Analysis, please see

- Wikipedia Page *Sentiment Analysis*
- Shashank Gupta's post on *Towards Data Science*

Data Resource

Data Audio waveforms for this project were obtained from the *Ryerson Audio-Visual Database of Emotional Speech and Song* (*RAVDESS*) dataset which is a dynamic, multimodal set of facial and vocal expressions by 24 professional actors (12 female, 12 male), vocalizing two lexically-matched statements in a neutral North American accent and can be found here: *RAVDESS*

For this project, we restricted our speech dataset to sample waveforms which included expressions of the following five(5) different emotions:

1. Calm
2. Happy
3. Sad
4. Angry
5. Fearful

Each expression is produced at two levels of emotional intensity (normal, strong), with an additional neutral expression. Thus, our dataset contains 1440 files: 60 trials per actor x 24 actors = 1440.

Original Analytic Approach

As part of their project, found here *GitHub*, researchers *Derek Hung* and *Mitesh Puthran* analyzed speech by building several machine learning models which attempted to detect emotions from the speech.

They investigated

- *Multilayer Perceptrons* (MLPs)
- *Long Short Term Memory* (LSTMs) models
- 1D *Convolution Neural Network* (CNN1D)

They concluded that both MLPs and LSTMs under-performed and that their CNN1D model performed best, so that will be our **baseline model**.

Waveform Analysis Package: They used the `librosa` library in `Python` to process and extract features from the audio files. Specifically, they used the `librosa` library to extract features *Mel Frequency Cepstral Coefficients* (**MFCC**s) from the waveforms. **MFCC**s are a feature widely used in automatic speech and speaker recognition and will be briefly described next.

Mel Frequency Cepstral Coefficients (**MFCC**s)

Human hearing is not equally sensitive in respect to different signal frequencies; in fact, humans are less sensitive to small differences at high frequencies than at low frequencies with the perceptual sensitivity being approximately logarithmic above a limit frequency of about 1000Hz.

A common approach is to model this behavior through a bank of triangular filters that are equally spaced on the *mel* scale (thus perceptually equidistant). That is, first few filters are *linearly* spaced, then remaining filters are *logarithmic* spaced for the higher frequencies.

They decided to use a bank of 216 triangular filters to produce the 216 **MFCC**s which become our **feature vector** to analyze. Therefore, the baseline model is a **CNN1D** that took the 216 **MFCC** feature vector as input and outputted a classification label into one of the 10 classes comprised of our 5 expressed emotions for both males and females.

Mel Frequency Cepstral Coefficients (**MFCC**s) [2]

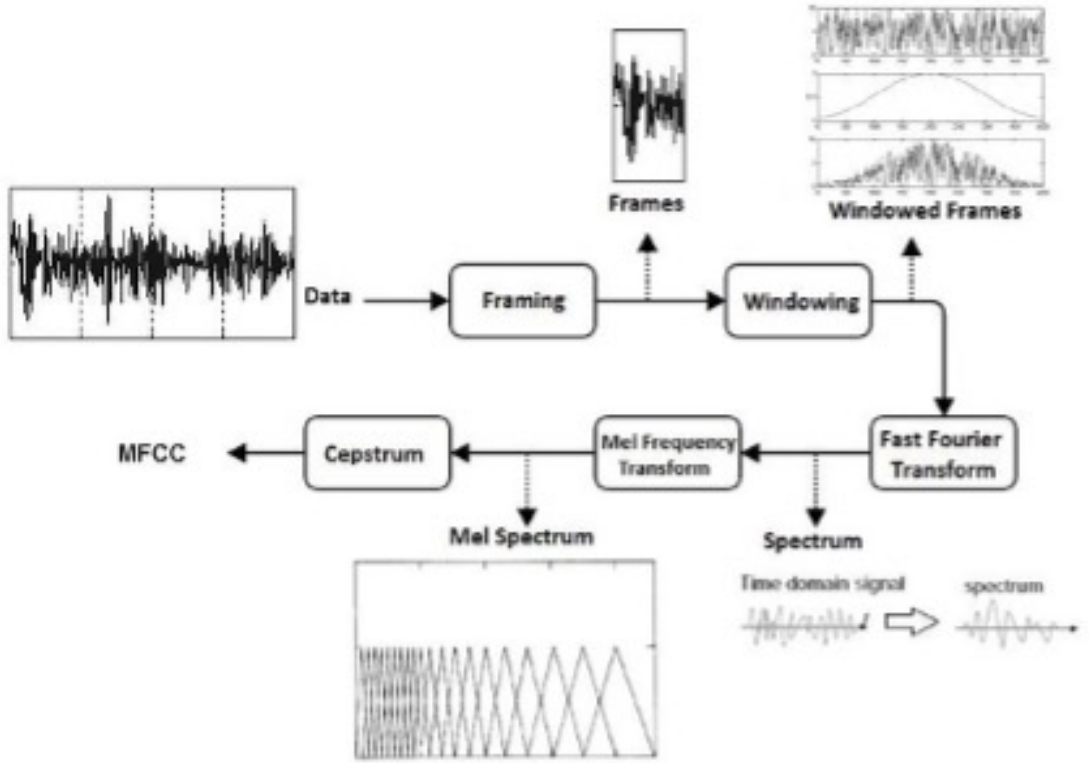

Figure 1: How *Mel Frequency Cepstral Coefficients* (**MFCC**s) are calculated

Mel Frequency Cepstral Coefficients (**MFCC**s) [3]

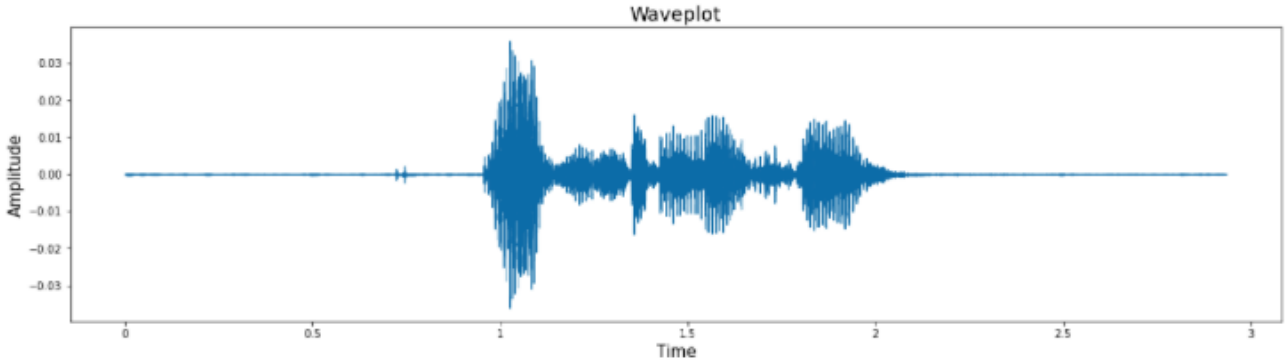

Figure 2: *Sample Waveform*

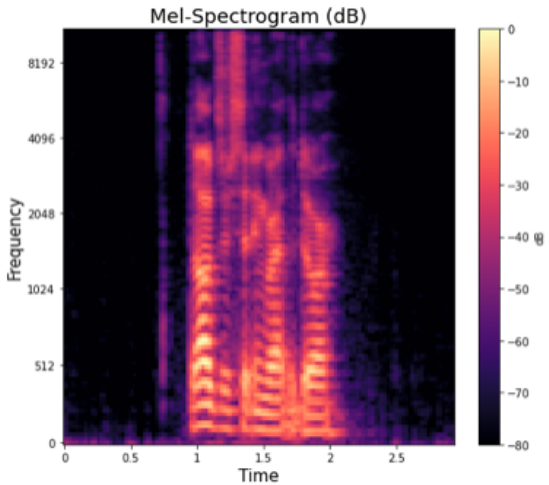

Figure 3: *Mel Spectrum* for Sample Waveform

Baseline Model: Architecture

Their baseline model is comprised of:

- 6 1D Convolutional layers, followed by *ReLU* activations, and 2 *Dropout* layers.
- There is one max pooling layer inserted between the second and third convolutional layer.
- They have a dense MLP with the number of classes followed by *softmax* activation as the final layer.
- The optimizer was *RMSProp* and the loss function was *CrossEntropy*.
- The *batch size* was set to 16 and the model was trained for 200 *epochs*.

Baseline Model: Architecture [2]

```
Model: "sequential_5"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d_27 (Conv1D)          (None, 216, 128)          768

 activation_32 (Activation)  (None, 216, 128)          0

 conv1d_28 (Conv1D)          (None, 216, 128)          82048

 activation_33 (Activation)  (None, 216, 128)          0

 dropout_8 (Dropout)         (None, 216, 128)          0

 max_pooling1d_6 (MaxPooling  (None, 27, 128)          0
 1D)

 conv1d_29 (Conv1D)          (None, 27, 128)           82048

 activation_34 (Activation)  (None, 27, 128)           0

 conv1d_30 (Conv1D)          (None, 27, 128)           82048

 activation_35 (Activation)  (None, 27, 128)           0

 conv1d_31 (Conv1D)          (None, 27, 128)           82048

 activation_36 (Activation)  (None, 27, 128)           0

 dropout_9 (Dropout)         (None, 27, 128)           0

 conv1d_32 (Conv1D)          (None, 27, 128)           82048

 activation_37 (Activation)  (None, 27, 128)           0

 flatten_5 (Flatten)         (None, 3456)              0

 dense_5 (Dense)             (None, 10)                34570

 activation_38 (Activation)  (None, 10)                0

=================================================================
Total params: 445,578
Trainable params: 445,578
Non-trainable params: 0
```

Figure 4: Baseline Model Architecture Summary

Baseline Model: Performance

The following graphs are typical of a training run for the Baseline model:

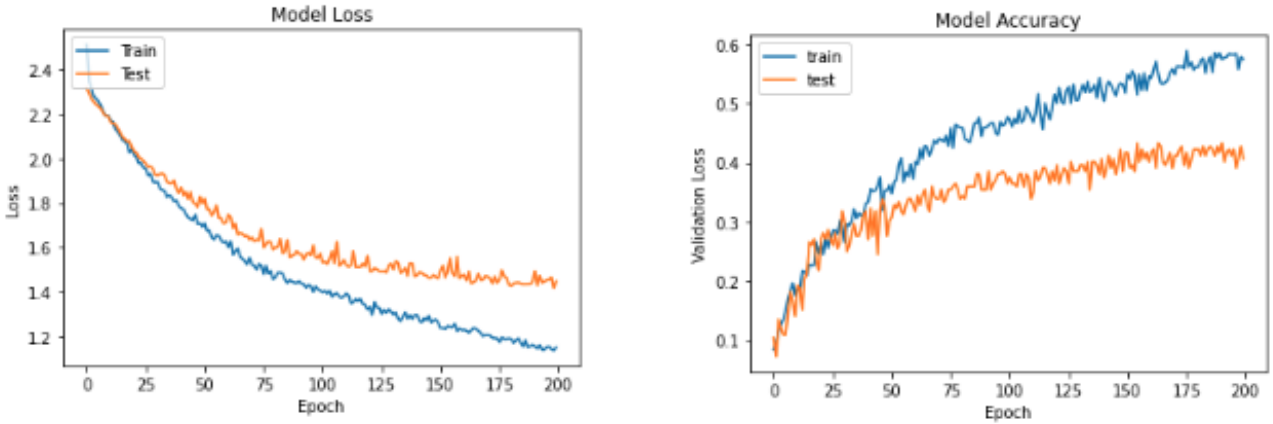

Figure 5: Performance of the Baseline Model: Left is *Loss*, Right is *Accuracy*

Although the researchers claimed to have achieved a validation accuracy of 70% with their preferred (baseline) model, our results showed a typical validation accuracy around 40%.

Proposed Model: Architecture

Our proposed model has been greatly reduced in size and is comprised of:

- 3 1D Convolutional layers, followed by *ReLU* activations and pooling layers, and 2 *Dropout* layers.
- The final dense MLP with the number of classes followed by *softmax* activation also is comprised of ℓ_1 *regularization* applied to both its *weight matrix* and *bias vector*.
- The optimizer was *RMSProp* and the loss function was *CrossEntropy*.
- The *batch size* was set to 16 and the model was trained for 200 *epochs*.

Proposed Model: Architecture [2]

Model: "functional_1"

Layer (type)	Output Shape	Param #
input_layer_1 (InputLayer)	(None, 1024, 1)	0
conv1d_3 (Conv1D)	(None, 993, 128)	4,224
max_pooling1d_2 (MaxPooling1D)	(None, 496, 128)	0
dropout_2 (Dropout)	(None, 496, 128)	0
conv1d_4 (Conv1D)	(None, 491, 128)	98,432
max_pooling1d_3 (MaxPooling1D)	(None, 245, 128)	0
dropout_3 (Dropout)	(None, 245, 128)	0
conv1d_5 (Conv1D)	(None, 240, 64)	49,216
global_average_pooling1d_1 (GlobalAveragePooling1D)	(None, 64)	0
dense (Dense)	(None, 10)	650

Total params: 152,522 (595.79 KB)

Trainable params: 152,522 (595.79 KB)

Non-trainable params: 0 (0.00 B)

Figure 6: Proposed Model Architecture Summary

Proposed Model: Performance

The following graphs are typical of a training run for the Proposed model:

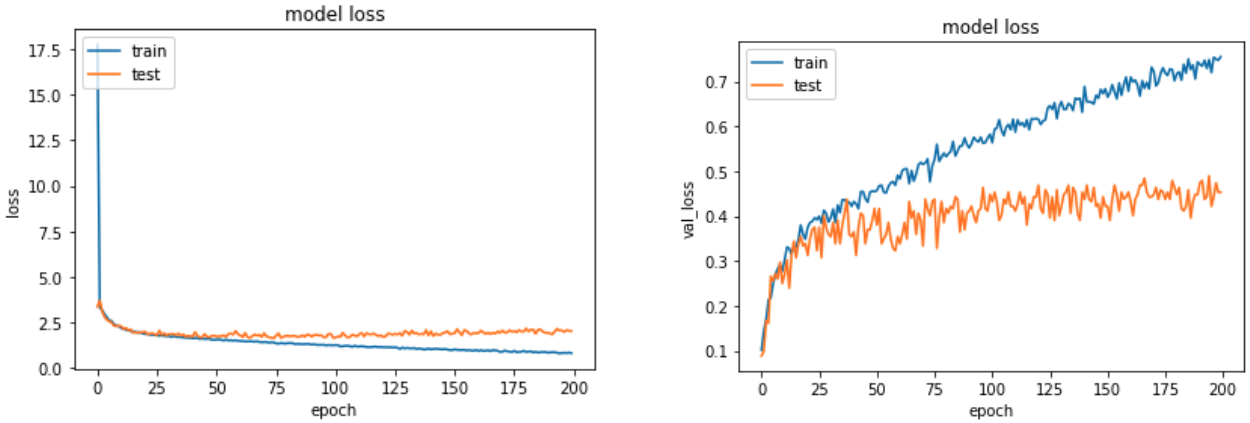

Figure 7: Performance of the Proposed Model: Left is *Loss*, Right is *Accuracy*

Project Summary

Findings While our Proposed model performed much better of the training data, this performance increase did not translate to any improvement of the generalization of our model as seen by the validation accuracy remaining relatively the same as the baseline model.

However, note that the Proposed model's overall size and complexity is substantially reduced from the Baseline model; the number of Baseline model coefficients is **445,578** while the Proposed model has only **152,522** model coefficients.

That is, the Proposed model provides similar results as the Baseline model but is only $\frac{1}{3}$ its size!

Future Work Investigate how to analyze the Mel Spectrum with 2D CNNs

References

- Wikipedia Page *Sentiment Analysis*
- Shashank Gupta's post on Towards Data Science: *Towards Data Science*
- Francois Chollet's Textbook *Deep Learning with Python, 2nd Ed.*, Manning, Shelter Island, NY, 2021
- *Ryerson Audio-Visual Database of Emotional Speech and Song* (*RAVDESS*) dataset found here: *RAVDESS*
- Researchers *Derek Hung* and *Mitesh Puthran* *GitHub* page.

Back-up Slide

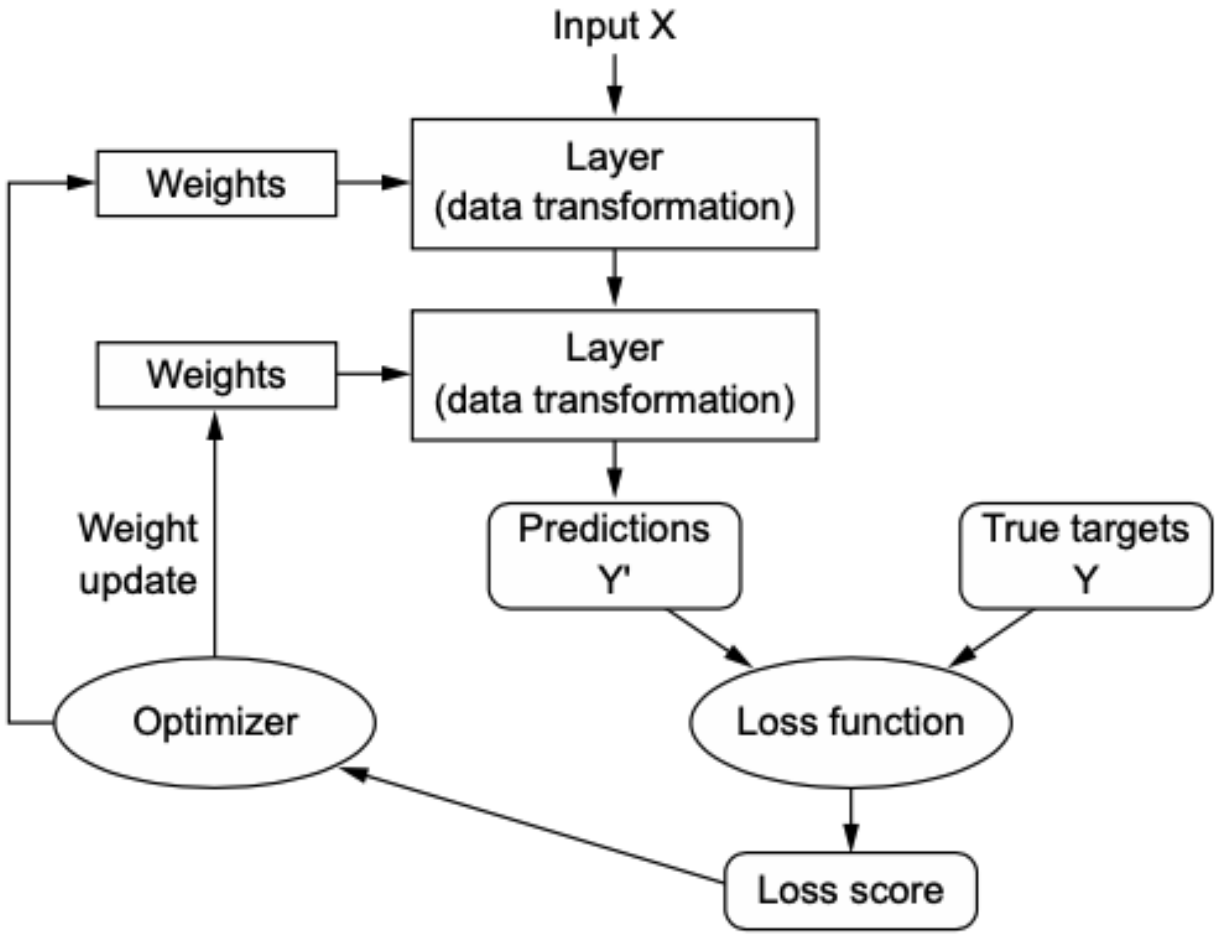

Figure 8: Relationships between Network Layers, Loss Function, and Optimizer. [see Chollet (2021)]