Non-intrusive Speech Quality Assessment with Convolutional Neural Networks and Audio Spectrograms

December 17, 2017

Jeff Cui

1 Summary

Non-intrusive speech quality assessment is widely used for monitoring the communication quality of lossy channels. In this final project, this problem is approached with recent advances in image processing and deep learning. The audio signal to be assessed is first cut into short, one-second-long segments. For each segment, a spectrogram is constructed by Fast Fourier Transform to represent the signal. Image features are then extracted using a pre-trained VGG16 ConvNet and used to predict whether the audio segment contains speech. If speech is detected, a speech quality opinion score from 1 (completely unintelligible) to 5 (clear speech) is predicted for the audio segment.

With a limited amount of training data, the speech detector was able to differentiate between noise and speech and achieve an accuracy of 91.6% on previously unseen data; the speech quality opinion score predictor was able to achieve a Mean Squared Error of 1.6 on previously unseen data.

To demonstrate knowledge of CS154 course content, here I also describe feature extraction and heuristics with FFT.

2 An FFT Feature-engineering Approach

Several speech features from Grancharov et al. (2006) is implemented as an alternative to convolutional neural network speech detection. Fast Fourier Transform was done on the audio signal, and these following features were calculated: **[?]**

- Spectral flatness:

$$\frac{\exp(\frac{1}{2\pi}\int_{-\pi}^{\pi}\log(P_n(\omega))d\omega)}{\frac{1}{2\pi}\int_{-\pi}^{\pi}P_n(\omega)d\omega}$$

where n is the audio frame index, and P_n the power spectrum. This is used to distinguish noise from speech. Human speech is varied along the power spectrum, while white noise tends to be flatter.

- Dominant frequency: $\text{argmax}(P_n)$

It is used as a heuristic for detecting human speech: if the dominant frequency is within 200–3000 Hz, it is more likely that the signal is human speech.

- Spectral centroid:

$$\frac{\int_{-\pi}^{\pi} \omega \log(P_n(\omega)) d\omega)}{\int_{-\pi}^{\pi} \log(P_n(\omega)) d\omega}$$

It is used to determine the most energetic area of frequency.

- Short-term energy:

$$\int_{-\pi}^{\pi} P_n(\omega) d\omega$$

This is the volume of the signal at frame n. It is used as a heuristic to detect unintelligibility due to low audio.

The features are thresholded and a simple majority vote is calculated to predict whether an audio frame contains audible speech.

2.1 FFT Frequency Bins

Fast Fourier Transform is a fast implementation of discrete Fourier Transform.

The continuous Fourier transform is

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x) e^{-i \cdot 2\pi \cdot x \cdot \xi} dx$$

The discrete Fourier transform becomes

$$\mathcal{F}(x) = \sum_{n=0}^{N-1} x_n \cdot e^{-i \cdot 2\pi \cdot \frac{n}{N} \cdot k}$$

Therefore, for each bin, the corresponding frequency is $n \cdot \frac{r}{N}$, where r is the sample rate, N is the number of samples, and n the index of the bin. This enables us to numerically calculate the features above.

2.2 Difficult Failure Cases

- Choppy audio. Choppy audio sounds like speech at each given time frame, and can trigger all the votes, but is unintelligible.
- Deceptive background noise. Loud or spectrally characteristic background noise can be mistaken as audible speech. Some failure examples from the experiment include: car engine noise, audible but unintelligible background conversations, and electric fan noise within the frequency range of human speech.

3 Alternative Approach: ConvNet

3.1 Hypothesis

While Fourier analysis has been widely applied to obtain a frequency domain representation for a given frame (typically 0.02 seconds) of audio that is richer than the raw signal, one short frame of audio cannot give us much information about the intelligibility of human speech because typical human utterances are far longer than 0.02 seconds.

To tackle this problem, a model has to take into account the temporal change of a signal. In the last decade, work on speech quality assessment or voice activity detection is typically done with limited consideration in this aspect. The paper discussed above, for example, only considers these features and their first-order derivatives. This is limited as it only considers change within two frames (0.04 seconds).

To perform signal analysis on the timescale of typical human utterances, we can use convolutions on audio spectrograms to extract features. This is a reasonable hypothesis because on an audio spectrogram, adjacent pixels represent adjacent time or frequency, and pixel brightness represent power. Audio signal is well-behaved, and the power between discrete adjacent frames or frequencies is often correlated. Convolutions and convolutional neural networks have been shown to handle quite well signals with such properties (e.g. photos).

Recent advances in image processing gives us convolutional neural networks, which are well-suited for this task. Typically, a convolutional neural network convolves the raw signal in increasingly larger strides: at first, the signal is convolved pixel-by-pixel with a small kernel to generate local features. Then, the features are aggregated with a max-pooling layer, and a larger convolution kernel is convolved against the processed signal at a larger stride. In the end, this enables a convolutional neural network to be aware of large patches of data. With a spectrogram as input, we can use it to analyze audio segments on a timescale that reflects human utterances.

3.2 Data

Ground truth data is collected by manually scoring 1238 1-second audio segments. If a segment has no speech, a score of 0 is assigned. If it has speech, an opinion score of 1 (completely inaudible) to 5 (clearly audible) is assigned.

I would like to thank Minerva Schools for providing the speech dataset, and Qiusu Wang for assisting with ground-truthing the data.

3.3 Algorithm

3.3.1 Feature Extraction

A Fast Fourier Transform is used to transform the raw audio signal to the frequency domain and construct a spectrogram. The brightness of the spectrogram corresponds to the signal magnitude at $(\text{time}, \text{frequency})$.

The FFT and sample configurations are: * Sample rate 44100Hz * Duration 1.137824s (50178 samples) * FFT Frame length 0.02s

This results in 129 pixels on the frequency axis from 0Hz to 22050Hz, and 223 pixels on the time axis.

3.3.2 Preprocessing: Input Dimensions

The pre-trained VGG16 ConvNet takes an input of shape (3, 224, 224), a BGR-coded square image. Our image is of shape (1, 129, 223).

Obviously in image processing, we would interpolate the image to a larger dimension by bicubic interpolation to reshape the input picture to suit the model. Although we arguably do not lose information by interpolation, it changes the original pixel values and complicates the problem. Therefore, the original inputs were simply kept in place, and black pixels were padded above 22050Hz, and beyond the duration of the signal. Then, the input is converted to BGR channels by copying the grayscale image brightness to each of the channels. This reshapes our input signal to (3, 224, 224) and it is now suitable for VGG16.

3.3.3 ConvNet Model

The pre-trained VGG16 image classification ConvNet is repurposed here for spectrogram analysis. It contains hidden convolutional layers that extracts features from an image. The last layer of the pre-trained VGG16 is a fully connected layer with softmax activation functions, used to classify images to one of the 1000 categories.

For the speech detection ConvNet, the last layer is discarded and replaced with 4 more layers:
Flatten
Fully connected (20), ReLU activation
Dropout (0.2)
Fully connected (2), Softmax activation

For the speech quality assessment opinion score regression ConvNet, the last layer is discarded and replaced with 3 more layers:
Flatten
Fully connected (30), ReLU actiation
Fully connected (1), Softmax activation

3.3.4 Discussion: Activation Functions

To demonstrate my CS154 course knowledge, here I briefly discuss activation functions and dead neurons in stochastic gradient descent. A recent advance in machine learning is the Leaky ReLU activation function to avoid dead neurons.

Performing stochastic gradient descent on the loss function requires that we calculate the gradient on the activation function.

The traditional Linear Unit activation function is

$$f(x) = \begin{cases} 0 \ (x < 0) \\ x \ (x >= 0) \end{cases}$$

which,

$$\nabla f(x) = \begin{cases} 0 \ (x < 0) \\ 1 \ (x >= 0) \end{cases}$$

meaning that if the weighted sum of inputs $\sum w_i y_i < 0$, we get a "dead neuron" because the gradient for this neuron $\nabla f(\sum w_i y_i)$ will always be 0 and it will not update its parameter values in stochastic gradient descent.

Leaky ReLU fixes this problem by having a small slope on the negative side:

$$f(x) = \begin{cases} 0.01x \ (x < 0) \\ x \ (x >= 0) \end{cases}$$

Although training for a neuron with this activation function is still slower if the weighted sum of inputs is negative, it will not stop completely during stochastic gradient descent.

3.3.5 Training

Due to the small (in deep learning terms) dataset size and computational resource and time constraints, the convolutional layer weights are fixed and backpropagation was only performed on the last few customized layers. If more data (and time) were available, it would make sense to also perform backpropagation gradient descent on the convolutional layer weights to fine-tune the original VGG16 weights to perform better predictions on spectrograms.

The speech detector ConvNet was trained 50 epochs with a categorical cross-entropy loss function. The speech quality assessment regressor ConvNet was trained 100 epochs with a mean squared error loss function.

The original dataset was split 80% for training, and 20% for testing the models.

3.3.6 Result

The speech detector had a 91.6% accuracy (284/310) on previous unseen data and was able to differentiate between sophisticated noise and speech. For audio segments containing speech, the speech quality opinion score regressor had a mean squared error of 1.6. The error distribution is below.

Error	-4	-3	-2	-1	0	1	2	3
Count	2	4	23	65	55	21	9	1

We can see that most of the data (78.33%) has an error within ± 1. WIth a limited amount of training data, the model was able to predict audio intelligibility reasonably well.

3.3.7 Predicting Audio of Arbitrary Length

The model was trained on one-second-long audio segments. To use the model on audio signals of arbitrary length, a rolling frame of 1 second, with an overlap of 0.5 seconds, can be used to score the audio. If the audio is shorter than the sample window, it can be zero-padded at the end.

3.3.8 Code

Code and model weights are available here:
https://drive.google.com/file/d/17betBdOH7ZLSZlcm2CZIaMmf2pIBSnlN/view?usp=sharing

4 Application

This project is motivated by the video interview part in Minerva Schools' admission process. Sometimes, applicants can have overwhelming background noise or connection problems during

interview, which corrupts their speech and makes it inaudible or barely audible. Currently, corrupt or noisy audio is only detected in preprocessing, when an admission processor listens to the audio stream. Then, if necessary, the applicant has to be notified to redo the interview. This could take two weeks or more. An automated speech quality assessment algorithm would significantly reduce the turnaround time and partially relieve the repetitive labor of human preprocessing.

5 Further Research

Further research can be done to improve predictive performance and lower time complexity. There are three directions to work on.

Expand the dataset. Because of time constraints, I was only able to manually score 1238 audio segments. This proves to be sufficient to train a usefully accurate model for speech opinion scores. A Mean Squared Error of 1.6 means that it gives scores that are no more than ± 1 away from the ground truth. An expanded dataset with 5000 samples can improve the model's predictive performance.

Compress the model. On an NVIDIA Tesla K80, the model took 6 seconds to evaluate 115 seconds of audio. This is because

Fine-tune VGG16 weights. The VGG16 convolutional layer weights are fixed and directly used to extract features from the spectrograms. With significantly more training data, the VGG16 weights can also be fine-tuned with gradient descent.

References

Wyse, L. (2017). Audio spectrogram representations for processing with Convolutional Neural Networks. *arXiv preprint arXiv:1706.09559*.

Grancharov, V., Zhao, D. Y., Lindblom, J., & Kleijn, W. B. (2006, September). Non-intrusive speech quality assessment with low computational complexity. In *INTERSPEECH*.