Intelligent Voice Identification with Neural Networks

CS 230 Winter 2018 Project Milestone
Category: Speech Recognition

Lloyd Maza `lmaza` | Michael Thompson `mt1` | David Troner `dtroner`

Introduction

In the modern era, automated speech recognition systems have become commonplace in a broad range of technology domains. While considerable attention is devoted to challenging tasks like speech-to-text transcription or voice modulation, we intend to focus on the slightly simpler problem of identifying speakers based on their voice. The ability to distinguish a speaker's identity among a group of people has considerable practical use, including automated air traffic control systems, identity verification for over-the-phone bank transactions, and speaker diarization for speech-to-text applications.
For the purposes of this project, we will look at the task of distinguishing a celebrity's voice from among a small group of about 20 speakers. This is a well-studied problem with a multitude of different known solutions, so we intend to leverage as much insight as possible from preceding research. Upon implementing a baseline voice identification system, we will devote considerable time to refining the model in order to maximize performance in the context of our specific problem.
This report outlines the current trajectory of the project, along with documenting the preliminary phase of implementing the system. Future directions for the project are proposed at the end.

Dataset

As mentioned earlier, the goal of this project is to distinguish the voices of around 20 people, hence the dataset will have to at least some audio of each person speaking. To make training easier, the dataset will be comprised of audio clips of each person speaking individually with no background voices or noise. For example, a training audio clip might have either Donald Trump or Barack Obama giving a speech, but the applause or crowd noise would be excluded. As expected, the audio clips will be labeled according to who is speaking; no text transcripts will be used.
The duration of audio for each training example is usually considered a hyperparameter of the neural network. Some typical durations for each training example are 1 second for Lukic et al. [2], 0.8 seconds for Torf et al. [3], and 2.5 seconds for Ge et al. [1]. As the planned architecture for our neural network most closely resembles the network used in Ge et al., each of our training examples will be 2.5 seconds in duration, although this might be optimized later in the design process. The initial audio will be stored in a mp3 format, and then pre-processed as described in the feature extraction section of this report.
The amount of data used in training also varies depending on the network implementation. Instances where extremely large datasets were used include Li et al. of the Baidu company [4]. This network trained on about 12,600 hours of voice data to classify about 250,000

speakers. This is gives an average of about 3 minutes of audio per speaker. This network was designed to do speaker classification as well as many other tasks which we do not plan on replicating. Relatively smaller datasets include Lukic et al. [2] and Ge et al. [1], both networks used about 25 seconds of audio per speaker for training to classify 100 and 326 speakers respectively. We plan on gathering 10 minutes of audio per speaker, more than all the previously mentioned neural networks. Because we plan on only classifying 20 speakers, we can use much more audio per speaker without excessive training times, hopefully allowing us to achieve very high accuracy. Additionally, gathering 10 minutes of audio per speaker will increase the overall size of our dataset, which has a risk of being too small due to the low number of speakers being classified. Approximately 97% of our data will be used for training, with 1.5% used for the development set, and 1.5% used for the test set. Lukic et al. used about 80% of their dataset for training, leaving approximately 5 seconds of audio per speaker for the dev set [2]. Our dataset breakdown gives 10 seconds of audio per speaker in both the test and dev set, and uses all remaining data for training. Gathering the necessary audio clips will be done by downloading audio from YouTube videos. Downloading audio clips from YouTube gives us precise control over the quality of the dataset, and flexibility in which people we want to identify. Given the ease of downloading YouTube audio, gathering 10 minutes of audio per speaker seems feasible. If we need more data than we can download from YouTube, an open-source speech data set will be used for training. There are many viable open source datasets downloadable for free online, some examples include compilations of labeled TED talks, audiobooks from multiple readers, and Google AudioSet. Again, these datasets are a backup plan in the case that manually downloading YouTube audio becomes infeasible.
So far, audio data has been collected from United States Presidents Donald Trump and Barack Obama. The final list of 20 people has not been decided, however some ideas for additional people include more politicians such as George Bush, Hilary Clinton, or Bernie Sanders; talk show hosts such as Jimmy Fallon; actors such as Tom Cruise; and of course, our own voices.

Approach

For the purposes of implementing this system, we intend to follow the basic approach outlined by Ge et al. in their paper on neural network-based speaker classification [1]. In their paper, they demonstrate the efficacy of a standard multi-layer perceptron in identifying a single person's voice from a finite group of speakers. While their approach discriminates among a large group ($\sim$200 people), we will attempt to generalize it to a much smaller group. We believe that by collecting substantially more data per person, we should be able to achieve a similar level of performance.

Feature Extraction

Since the method we will be implementing makes use of standard neural network architecture, all audio data must be processed into a feature vector which can be fed into the input layer. To this end, we intend to follow the basic feature extraction method outlined by Ge et al. It should be noted that the same feature extraction process will be employed for both training and test sets to ensure that all data is drawn from the same distribution. First, the audio will be normalized to unit amplitude to ensure consistency across examples. Next, we will use voice activity detection (VAD) to reduce the audio so it only contains voiced speech. We intend to employ a high classification threshold in this step to make absolutely sure our training data consists entirely of human speech without irrelevant noise. Our literature review suggests that this will improve the performance of the model while reducing training time by removing unnecessary portions of the data.
The most important part of the data processing is the actual feature extraction which converts an audio file to a feature vector. For this purpose we will use Mel-frequency cepstral

coefficients (MFCCs), which are generally regarded as the de-facto standard feature set used in most speech recognition tasks. It should be noted that a drawback of MFCCs is that they eliminate voice-specific characteristics of the speech signal, so this may present a potential challenge for our voice identification system [2]. To get around this obstacle, we intend to normalize each speaker's MFCCs by his/her own mean and variance. This process should be useful in re-introducing some of the idiosyncrasies of each speaker's voice.

Network Architecture

The neural network arcitecture outlined in by Ge et al utilizes a single hidden layer of 200 nodes along with a sum of multiple binary classification losses at the output layer of 200 nodes. However, this was determined through a grid search of at most 2 hidden layers and 400 nodes. Our approach seeks to improve on this model through adding network depth beyond 2 hidden layers along with a softmax output layer and corresponding usage of categorical crossentropy loss. Our approach is implemented using Keras and incorperates L2 regularization and Adam optimization over mini batches to improve performance and computation time. Batch nomalization on each hidden layer is utilized. The nominal depth of the network is 5 hidden layers with ReLU activations, each with 390 nodes correspoding to the length of input feature vectors. However, the number of nodes per layer and number of layers are hyperparameters that will be tuned.
It is notable that we are opting to use a network consisting of fully-connected layers and not an RNN since the speaker recognition task is text-independent and therefore sequence data is not required. Another framework from our literature review incorperated utilizing a CNN on specrogram images created from audio segments, as demonstrated in [2]. We are investigating this arcitecture; however, with our relatively low dimensional input vectors and ease of data preprocessing, we are opting to focus on the fully-connected layer apporach as this yeilded high accuracy for speaker identification tasks. Much of the code architecture has been built with inputs for tuning hyperparameters. This code is attached at the end of this report, and has been sucessfully run on preliminary audio data of Presidents Trump and Obama yeilding 90% test accuracy.

Future Work

Future work largely centers around hyperparameter tuning of the neural network to improve performance along with refined data generation from audio clips. Key hyperparameters of the network we will be tuning are network depth, number of nodes, lambda from L2 regulation, mini-batch size, learning rate, along with investigating other regulation methods such as L1 and dropout. Future iterations of the dataset might include impressionists such a Frank Caliendo, or audio clips taken from the same person but separated by many years.

Contributions

All members of this team have contributed in equal amounts to the overall progress of the project. In particular, the task of literature review was shared among all three members. Since then, Michael has focused on developing the dataset and acquiring data, Lloyd has focused on data pre-processing and feature extraction, and David has focused on comparing network architectures and developing a preliminary implementation of the network.

Links to Current Code

`https://github.com/lloydmaza/CS230-FinalProject/tree/master/Code`
`https://colab.research.google.com/notebook#fileId=1mBJSx3KpnIIi5o-wW4iVxCk85fgKVXDE`

data loading will only run on Google Colab

References

[1] Z. Ge, A. Iyer, S. Cheluvaraja, R. Sundaram, A. Ganapathiraju, "Neural Network Based Speaker Classification and Verification Systems with Enhanced Features," in *Intelligent Systems Conference*, London, UK, September 7-8, 2017.

[2] Y. Lukic, C. Vogt, O. Dürr, T. Stadelmann, "Speaker Identification and Clustering Using Convolutional Neural Networks," in *IEEE International Workshop on Machine Learning for Signal Processing*, Salerno, Italy, September 13-16, 2016.

[3] A. Torfi, N. Nasrabadi, J. Dawson, "Text-Independent Speaker Verification Using 3D Convolutional Neural Networks," in *Computing Research Repository*, June 28. 2017.

[4] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, Z. Zhu, "Deep Speaker: an End-to-End Neural Speaker Embedding System," in *Computing Research Repository*, May 5, 2017.