UDC 004.934.8'1

SPEAKER RECOGNITION USING NEURAL NETWORKS

Lu Gangfan

group 267311

Belarusian State University of Informatics and Radioelectronics, Minsk, Belarus

Scientific supervisor: Petrov S.N., PhD, associate professor

Annotation. This project demonstrates how to build a speaker recognition system using deep learning techniques. The system uses MFCC to extract features from audio data and capture spectral and time domain information of speech. After comparing traditional classification methods and neural network classification methods, then chooses a recurrent neural network (RNNs) to process of sequence data using. The project was trained and evaluated on the famous audio dataset VoxCeleb1 to train and evaluate various speaker recognition models using python. The system achieved a test accuracy of 93%. This result demonstrates that the system is able to effectively distinguish between different speakers.

Keywords: Speaker Recognition. MFCC, Deep Learning. Recurrent Neural Networks

Introduction. Speaker recognition is a biometric technology that utilizes the voice characteristics of a speaker to verify or identify his or her identity. Traditional speaker recognition systems rely on manually created features and machine learning algorithms that can be limited in terms of accuracy and robustness. Deep neural networks can learn complex data representations and have achieved state-of-the-art results in various speaker recognition tasks.

In this paper, we introduce different classification methods and compare the limitations of traditional classification methods compared to neural networks. In recent years, deep learning has become a powerful tool for speech and audio processing tasks. Therefore, we propose a deep neural network-based speaker recognition system. In this system, the audio recordings are segmented into different segments, preprocessed to remove noise, features are extracted from the audio data by Mel Frequency Cepstrum Coefficients (MFCC), and then the sequence data is processed using Recurrent Neural Networks (RNN). We evaluated our system on the VoxCeleb1 dataset and tested it with 93% accuracy. Deep neural networks can learn complex representations of speech data and can be used to build robust and accurate speaker recognition systems.

Main Part. The collected speech information is first removed from the muted portion and then noise reduction is performed using adaptive filtering or spectral subtraction to facilitate further subsequent processing. MFCC stands for Mel-Frequency Cepstral Coefficients, and it is a technique for extracting features from audio signals that are useful for speech recognition and other tasks. Here is the MFCC process:

1. Pre-emphasis. The first step is to apply a pre-emphasis filter to the audio signal, which boosts the high-frequency components and reduces the effect of noise. The pre-emphasis filter is a first-order high-pass filter, and it can be expressed as:

$$y[n] = x[n] - \alpha x[n-1]$$

Where x[n]is the input signal, y[n]is the output signal, and α is a constant between 0.9 and 1.0.

2. Framing and windowing. The next step is to divide the signal into short frames of equal length, typically 20 to 40 milliseconds. This is done because the frequency spectrum of the signal changes over time, and we want to capture the local characteristics of the signal. The frames are usually overlapped by 50% to avoid discontinuities at the frame boundaries. Each frame is then multiplied by a window function, such as a Hamming window, to reduce the spectral leakage caused by the finite length of the frame. The window function can be defined as:

$$w[n] = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right)$$

Where w[n] is the window function, N is the frame length, and n is the sample index.

3. Fourier transform. The third step is to apply the discrete Fourier transform (DFT) to each frame to obtain the frequency spectrum of the signal. The DFT can be computed as:

$$X[k] = \sum_{n=0}^{N-1} x[n]w[n]e^{-j\frac{2\pi kn}{N}}$$

Where X[k] is the DFT of the frame, x[n]is the signal, w[n]is the window function, N is the frame length, k is the frequency index, and j is the imaginary unit.

4. Mel filter bank. The fourth step is to apply a set of triangular filters to the frequency spectrum to approximate the human perception of sound frequency. The filters are spaced according to the mel scale, which is a nonlinear scale that relates the perceived pitch of a sound to its actual frequency. The Mel scale can be defined as:

$$m = 2595\log_{10}\left(1 + \frac{f}{700}\right)$$

Where m is the Mel frequency, and f is the linear frequency. The filter bank consists of a number of filters, typically 20 to 40, that cover the entire frequency range. Each filter has a triangular shape, with a peak at the center frequency and zero values at the edges. The filter bank can be represented as a matrix, where each row corresponds to a filter and each column corresponds to a frequency bin. The filter bank can be applied to the spectrum by multiplying it element-wise and summing up the results. This produces a vector of filter bank energies, which represent the amount of energy in each filter.

5. Logarithm. The fifth step is to take the logarithm of the filter bank energies to compress the dynamic range and mimic the human loudness perception. The logarithm can be expressed as:

$$S[m] = \log\left(\sum_{k=0}^{K-1} H[m,k]\,|X[k]|^2\right)$$

Where S[m] is the log filter bank energy, H[m,k] is the filter bank matrix, X[k] is the spectrum, m is the filter index, k is the frequency index, and K is the number of frequency bins.

6. Discrete cosine transform. The final step is to apply the discrete cosine transform (DCT) to the log filter bank energies to obtain the MFCCs. The DCT can be computed as:

$$C[n] = \sqrt{\frac{2}{M}}\sum_{m=0}^{M-1} S[m]\cos\left(\frac{\pi n}{M}\left(m + \frac{1}{2}\right)\right)$$

Where C[n] is the MFCC, S[m] is the log filter bank energy, n is the cepstral index, m is the filter index, and M is the number of filters. The DCT reduces the correlation between the filter bank

energies and decorrelates the features. Usually, only the first 12 or 13 coefficients are kept, as they contain most of the relevant information. The MFCCs can also be augmented with the energy of the frame and the first and second derivatives of the MFCCs to capture the temporal dynamics of the signal.

A cornerstone in speech processing, MFCCs approximate the frequency resolution of the human auditory system. Through mathematical transformations applied to the frequency spectrum, MFCCs distill complex information into a compact yet expressive representation, crucial for nuanced speech pattern recognition. The resulting MFCC features are a compact representation of the speech signal that captures important information about the speaker's voice.

Feature extraction is the method and process of using computers to extract characteristic information in sound signals. During the speech recognition process, the text will be converted into a coded form and separated into syllables, phonemes, etc. Feature extraction is a crucial phase in the speech recognition process, involving the distillation of pertinent information from the preprocessed speech signal. This section delves into specific techniques employed to extract distinctive features, laying the groundwork for subsequent model training and recognition. Speaker recognition system whole workflow as Figure 1.

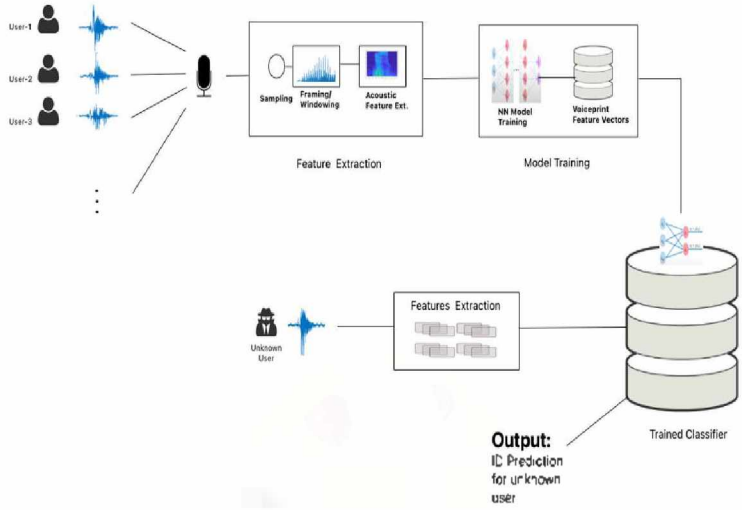

Figure 1 – Speaker recognition system workflow

Spectrum Analysis

Spectrum analysis is an important part of feature extraction, aiming at revealing the basic frequency characteristics inherent in speech signals. The process of MFCC feature extraction is as follows: firstly, the speech signal is divided into several segments according to the time; then, the fast Fourier transform is performed on each segment of the signal, and the optical spectrum can be obtained after the transformation; according to the energy envelope of the optical spectrum, the energy envelope is discretized, and a vector can be obtained. This vector is the MFCC vector.

And the peaks and modes in the spectrogram can represent specific frequencies or modes in the audio signal. For example, a formant in speech can be recognized as a concentration of energy at a specific frequency.

Temporal Feature

Determines the fundamental frequency of the speech signal, representing the perceived pitch of the speaker's voice. Pitch extraction aids in capturing intonation patterns and is valuable for recognizing emotions and nuances in speech.

Zero Crossing Rate: Quantifies the rate at which the speech signal crosses the zero-amplitude axis. This feature is useful for discerning voicing characteristics and is applied in various speech analysis tasks.

Statistcal Feature

Extract statistical measures such as mean, variance, skewness, and kurtosis from the speech signal. These measures offer insights into the distribution and characteristics of the signal.
Energy Computation: Compute the energy of the speech signal over different frequency bands. Energy features help in distinguishing between voiced and unvoiced segments of speech.
The combination of spectral, temporal, and statistical features creates a comprehensive representation of the speech signal. These features collectively capture the nuanced patterns and variations essential for accurate speech recognition. To enhance computational efficiency and reduce redundancy, dimensionality reduction, the technique of Principal Component Analysis (PCA) is applied to the feature set. This step retains the most relevant information while minimizing the number of features.
Effective feature extraction transforms raw speech signals into compact, informative representations, laying the groundwork for subsequent stages in the speech recognition process. These features serve as the input to models, such as Recurrent Neural Networks (RNNs), fostering accurate and efficient recognition of spoken language.
This system uses Librosa (a Python package for audio analysis) provide tools to analyze and visualize audio data, including functions for optimizing feature extraction, time-series representation, and visualization of audio signals. It is commonly used in the field of music information retrieval and audio signal processing.

Speech recognition technology classification

The first category is model matching methods, including vector quantization (VQ), dynamic time warping (DTW), etc;
The second category is probabilistic statistical methods, including Gaussian Mixture Model (GMM), Hidden Markov Model (HMM), etc.;
The third category is discriminator classification methods, such as support vector machine (SVM), recurrent neural network (RNN), deep neural network (DNN), etc., as well as various combination methods.
Speaker recognition is actually a process of encoding first and then decoding. Signal processing and feature extraction are the encoding process. In other words, it is a pattern recognition based on speech feature parameters. That is, through learning, the system can classify the input speech according to a certain pattern, and then find the best matching result based on the judgment criteria.

Neural networks

Recurrent Neural Networks (RNNs) are neural networks that process sequential data, such as speech or text. In speech recognition, RNNs can learn the contextual information of the speech to improve the accuracy of the recognition. The role of MFCC is to extract relevant features from the audio data and then feed these features into subsequent layers of the RNN for higher level processing and interpretation. This integration improves the overall performance of the speech recognition model. Because of its ability to model sequential dependencies and capture temporal patterns in the data, the RNN is well suited for some aspects of speaker recognition. RNNs can also be used to model the variability in speech patterns, which is essential for robust speaker recognition systems.
When defining the RNN model, ReLu and SoftMax are activation functions used in different layers of the neural network model. ReLU (Rectified Linear Unit) is an activation function commonly used in hidden layers of neural networks. Mathematically, it is defined as ($f(x) = \max (0, x)$), which means that the output is zero for negative inputs and equal to the input for positive inputs. ReLU helps introduce non-linearity to the model, allowing it to learn complex patterns.
SoftMax is an activation function used in the output layer of a neural network for multi-class classification problems. It converts the raw output scores (logits) into probability distributions over multiple classes. The output of the SoftMax function is a vector where each element represents the probability of the corresponding class. The class with the highest probability is predicted as the final output. In the below model architecture, the last layer uses SoftMax activation to obtain probability distributions over the different classes (speakers) for the final prediction. The preceding layer uses relu activation for introducing non-linearity in the hidden layer. The choice of activation

functions depends on the nature of the problem and the desired properties of the network.
What's more, different types of loss functions serve different purposes, and the choice often depends on the nature of the problem you are trying to solve, this system uses Binary Crossentropy, Categorical Crossentropy, Sparse Categorical Crossentropy to evaluate the result.

Conclusion. The proposed system implements a speaker recognition system based on Neural Networks using python. In experiments using the audio dataset VoxCeleb1 as the training set and test set, splitting the data into 70% train and the remaining will be split equally into validation and test datasets. The training parameters are set to learn at a rate of 0.001, and the optimizer uses Adam.

The training process monitors the classification accuracy and loss function changes. The project used accuracy, precision, recall, and F1-score as evaluation metrics. The results of the project showed that the neural network model performed the best on all metrics, achieving 0.95 accuracy, 0.93 precision, 0.95 recall, and 0.98 F1-score. The test results on the validation set can reach nearly 90% accuracy. In summary, the system achieved a test accuracy of 93% on the VoxCeleb1 dataset. It is shown that the use of neural networks combined with MFCC features is effective for speaker recognition tasks.

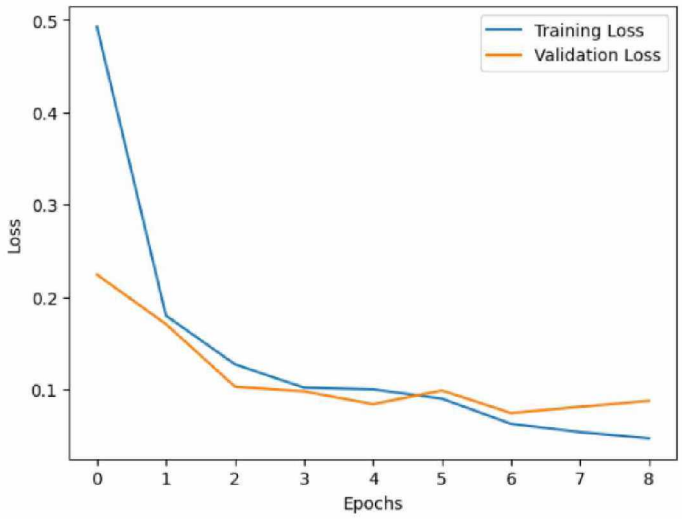

Figure 2 – Speaker recognition implementation and train results

Of course, it has many shortcomings and needs to deal with the effects of multiple factors, such as noise, accent, speech rate, intonation, and context. These factors can lead to changes and interference in the speech signal, reducing the accuracy and robustness of speech recognition. Therefore, algorithms and techniques for speech recognition need to be continuously optimized and improved to adapt to different speech scenarios and requirements.

References

1. From Natural to Artificial Intelligence - Algorithms and Applications Edited by Ricardo Lopez-Ruiz

2. D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," Digital Signal Processing,

3. J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep speaker recognition," Proc. Interspeech 2018, 2018, pp. 1086-1090.

4. Research on Speaker Recognition base on the Ensemble Deep Learning Model, Dewei Huang

5. Research on the Method of Voiceprint Recognition Based on Deep Neural Network, Pu Lili

УДК 004.934.8'1

РАСПОЗНАВАНИЕ ДИКТОРА С ИСПОЛЬЗОВАНИЕМ НЕЙРОННЫХ СЕТЕЙ

Лу Гангфань

гр. 267311

Белорусский государственный университет информатики и радиоэлектроники, г. Минск, Республика Беларусь

Научный руководитель: Петров С.Н. – к.т.н., доцент

Аннотация. Показан подход к построению системы распознавания диктора с использованием методов глубокого обучения. Система использует мел-частотные кепстральные коэффициенты в качестве характеристик аудиоданных. Проведено сравнение традиционных методов классификации и классификации с использованием нейронных сетей, по результатам сравнения для обработки речевых сигналов выбраны рекуррентные нейронные сети (RNNs). Модель, реализованная на языке программирования Python, была обучена на датасете VoxCeleb1. Точность распознавания (accuracy) составила 93%, что позволяет модели эффективно распознавать различных дикторов.

Ключевые слова: распознавание диктора, MFCC, глубокое обучение, рекуррентные нейронные сети