Speaker Classification with Deep Learning

Nathan Bonavia Zammit

nathan.bonavia.20@um.edu.mt

Abstract

The purpose of this project was to develop a speaker identification system using deep learning models on a provided corpus of speakers. The system aims to accurately identify the speaker from a given audio sample. This report summarizes the implementation and performance of the developed speaker identification system.

1. Introduction

The purpose of this report is to document the SID (Speaker Identification) system built for a given corpus of speakers using Deep Learning Models. The task of speaker identification involves the use of machine learning algorithms to recognize the speaker in an audio signal based on the unique characteristics of their voice. In this report, we will describe the methods and techniques used to build the SID system, including data preparation, model selection, and evaluation metrics.

Speaker identification has numerous applications, ranging from forensic analysis, speech recognition, and speaker diarization to name a few. In recent years, Deep Learning Models have shown promising results in speaker identification tasks. These models are designed to learn complex relationships between inputs and outputs by building a hierarchy of representations through multiple layers. As a result, they have the ability to capture high-level abstractions in audio signals, making them a suitable choice for speaker identification tasks.

The corpus of speakers used for this assignment was preselected and provided for the purpose of building the SID system. The corpus consisted of recordings of multiple speakers, each with a different voice and speaking style. The recordings were made in different environments and under various conditions, making the task of speaker identification more challenging. The goal of this assignment was to build an SID system that could accurately identify the speakers in the provided corpus.

To achieve this goal, several steps were taken to preprocess the audio data and prepare it for use in the Deep Learning Models. This included normalizing the audio signal, removing background noise, and extracting relevant features. Next, multiple Deep Learning Models were trained and evaluated using the prepared data. The selection of the final model was based on the accuracy and performance of the models on the validation set.

In this report, we will describe the methods used to preprocess the data and build the SID system in detail. We will also present the results of the evaluation of the final model and discuss the performance of the system. Finally, we will conclude with a discussion of the strengths and limitations of the system and suggest potential avenues for future work.

2. Architecture

The architecture of the speaker identification system is a crucial aspect of the project, as it defines how the system processes audio signals to accurately identify the speaker. The goal of this section is to provide a comprehensive overview of the architecture of the system and its various components.

2.1. Data Preparation & Extraction

The first step in building the speaker identification system is to extract audio features from the recordings. The recordings in the corpus are first loaded into the system using the librosa library. The audio signals are then transformed into Mel-frequency spectrograms, which are two-dimensional representations of the spectral content of audio signals. The spectrograms are created using the librosa feature extraction library and the audio signals are resampled to 16,000 samples per second, which is a common sampling rate for speech signals. Finally, the MFCCs are extracted from the Mel-spectrograms using the librosa library. The MFCCs are a commonly used feature representation for speech signals, and they capture the spectral envelope of the speech signals, which is a crucial factor in speaker identification.

2.2. Data Splitting

The extracted MFCCs are then divided into training, validation, and testing sets. The training set is used to train a deep neural network model, which is implemented using the TensorFlow library. The validation set is used to evaluate the model's performance during training and to prevent overfitting. The testing set is used to evaluate the final performance of the trained model. The labels for the training, validation, and testing sets are obtained by encoding the speaker names using the scikit-learn LabelEncoder class.

2.3. Model Training

The neural network model consists of several fully-connected layers, each with a rectified linear unit (ReLU) activation function. The model is trained using the categorical cross-entropy loss function and the Adam optimization algorithm. The training process is monitored using the accuracy metric, which measures the fraction of correct predictions made by the model. After training, the model is evaluated using the validation set, and the results are used to determine the optimal number of training epochs.

The evaluation of the training process of the model showed that it reached 100% accuracy on the 63rd epoch. However, the validation accuracy was not improved after 100 epochs of training, which indicates that overfitting may have occurred. Overfitting is a common issue in machine learning where the model becomes too specialized to the training data and performs poorly on unseen data.

In this case, the model may have memorized the training data and was not able to generalize well to new data. This could be due to having too many epochs, not having enough diverse data for validation, or having an overly complex model.

3. F-Score Evaluation

F-Score is a commonly used evaluation metric in the field of speaker identification. It is a combination of precision and recall, which are two measures of the accuracy of a model's predictions. The F-Score provides a balanced assessment of the performance of the model by considering both the false negatives and false positives.

In speaker identification, the goal is to determine the speaker of an audio clip. The F-Score can be used to evaluate the performance of the model by comparing its predictions to the ground truth (i.e. the actual speaker of the clip).

To calculate the F-Score, we first compute the precision, which is the number of true positive predictions divided by the total number of positive predictions. Precision measures the accuracy of positive predictions, i.e. how often the model correctly identifies the speaker.

Next, we compute the recall, which is the number of true positive predictions divided by the total number of actual positive cases. Recall measures the completeness of the positive predictions, i.e. how often the model identifies all of the speakers.

Finally, the F-Score is calculated as the harmonic mean of precision and recall. It ranges from 0 to 1, with a score of 1 indicating perfect accuracy.

By using the F-Score in speaker identification, we can get a more comprehensive evaluation of the model's performance. For example, a model with a high precision but low recall may have a high accuracy in positive predictions, but it may miss many actual speakers. On the other hand, a model with high recall but low precision may identify many speakers, but it may also have a high number of false positive predictions.

4. Future Works

In the present work, we built a speaker identification model and found that there was an overfitting issue present in our model, as shown by the high accuracy achieved on the training data but low accuracy on the validation data. There are several techniques that can be employed to mitigate the overfitting problem and improve the speaker identification model.

Early Stopping: Early stopping is a technique where training is stopped once a certain criteria is met, such as no improvement in the validation loss after a certain number of epochs. This helps to prevent overfitting by stopping the training when it starts to over-specialize on the training data.

Dropout: Dropout is a regularization technique where, during training, a random subset of the neurons in the network are disabled. This helps to prevent overfitting by forcing the network to learn more robust representations and reduce the dependence on any individual feature.

Regularization: Regularization is another technique that helps to prevent overfitting by adding a penalty term to the loss function that penalizes certain forms of overfitting. The most common forms of regularization are L1 and L2 regularization, which add a penalty term proportional to the absolute or square value of the parameters, respectively.

By implementing Early Stopping, Dropout, and Regularization techniques, we could have improved the results of the speaker identification model. In future works, it would be interesting to explore the effect of these techniques and their combination on the performance of the speaker identification model. Additionally, we could also consider exploring other deep learning architectures and experimenting with different sets of hyperparameters to see if they yield better results.

5. Conclusions

In conclusion, the training process of the Speaker Identification model showed promising results, with 100% accuracy being achieved on the 63rd epoch. However, the validation accuracy was not improved whatsoever, which suggests overfitting problems. To tackle this issue, several techniques could be applied in future works, including Early Stopping, Dropout, and Regularization. These techniques can help prevent overfitting by controlling the model's complexity, avoiding memorization of the training data, and regularizing the model parameters. Additionally, the use of F-score as an evaluation metric could also be applied to further improve the results of the model. By incorporating these improvements, we can continue to enhance the accuracy and robustness of the Speaker Identification system and achieve better performance in real-world applications.