Audio File Classification

Farhad Yousefi Razin
Politecnico di Torino
310027
farhadyousefirazin@gmail.com

Mohamad Samaei
Politecnico di Torino
314577
samaeii.mohamad@gmail.com

***Abstract*—In this report we present an attitude toward Audio File Classification problem. More specifically, audio files are trimmed and padded to the longest audio length and then the MFCC of each audio file is extracted. After balancing the data set through oversampling and standardization of training data, we use two alternative approaches to train the model, the first of which leads to better results. The first one, training the model by means of standardized MFCCs in Artificial Neural Networks, shows a margin of increased accuracy compared to other solution.**

***Index Terms*—component, formatting, style, styling, insert**

I. Problem Overview

This project is a classification problem on a number of .wav audio files, the details of which are described in a CSV file containing some information about the speaker such as gender and age range.

In order to correctly identify the intent of the speaker, we must identify two elements: action and object. Every audio labeled with a specific class has a different way to express its intended purpose. For instance, speakers use various phrases to state the label ”decrease volume”, hence the goal of the project. The data set is divided into two parts:

- There are 9,855 recordings in development set with various details, objects, and actions.
- An evaluation set, comprised of 1455 recordings.

We will need to use the development data to build a classification model to correctly label the audio files in the evaluation data set.

It is worth mentioning that the data set is not balanced. This issue is dealt with by the process of oversampling in which a number of data points are chosen and duplicated to make equal frequency for each label.

The audio files are of varying lengths. Figure 1 shows the distribution of their lengths. Evidently, a limited number of data deviate from the norm. These outliers are not deleted to avoid data elimination. However, this problem is addressed in data preprocessing.

Figure 2 represents the time-amplitude plot of a random file, which shows the changes in volume over time. Figure 3 and Figure 4 plot a random audio file which scales the energy of a range of frequencies (the first on a linear scale and the second on a logarithmic scale). According to Figure 3, some frequencies have higher energy than others.

The logarithmic scale (Figure 4) is used since the human ear perceives sound in a logarithmic fashion, with the magnitude of sound perceived being proportional to the logarithm of its physical intensity. In general, the curve will show peaks and valleys, with the peaks representing the most prominent frequencies in the audio signal, and the valleys representing the frequencies with the least energy.

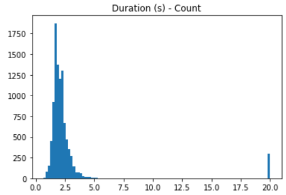

Fig. 1. Duration of recordings

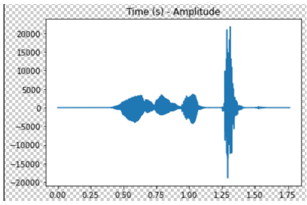

Fig. 2. Time-Amplitude Curve

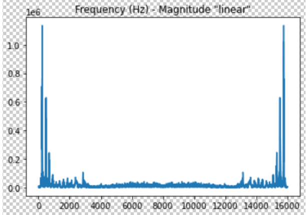

Fig. 3. Magnitude of a recording on linear scale

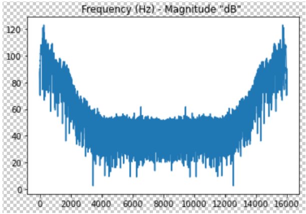

Fig. 4. Magnitude of a recording on logarithmic scale

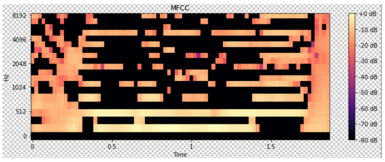

Fig. 5. Mel-Frequency Cepstral Coefficients (MFCC)

II. Research Approach

For the approach used in this project, a classification pipeline is followed:

A. Data Preprocessing

Initially, we remove any leading and trailing silence from the signal, a process called trimming. This procedure is beneficial in preprocessing as it makes training the more smooth and yields improved results by removing irrelevant or distracting content from the audio signal. In the second step, we use padding in order to make fixed-size input variables.

Afterwards, to obtain a compact and informative representation of the spectral characteristics of the signal, we extract Mel-Frequency Cepstral Coefficients (MFCC) for each audio file. Figure 5 represents a sample MFCC for an audio file of the data set.

MFCCs are numerical coefficients that represent a set of features extracted from an audio signal. The MFCC coefficients are calculated by converting the audio signal from the time domain to the frequency domain, then representing it as a set of coefficients on a log-metric scale that capture the spectral envelope of the sound [1]. The logarithmic scale is used to better match the non-linear perception of human hearing.

Oversampling the development data set is the next step. To reach the majority class quantity, data samples from minority classes are replicated.

Ultimately, both training data and test data is standardized. Standardization of the features is a common preprocessing step to ensure that each feature has the same scale. For this purpose, the mean and standard deviation of each set are considered individually not interchangeably.

B. Selection of Model

In this project two approaches are tested:

Approach 1: Upon finishing preprocessing, there is a standardized MFCC for each audio file which is then fed to an Artificial Neural Network. ANNs are interconnected nodes organized into layers. The input layer receives data, which is then processed and transformed by the hidden layers and an output layer produces the final result. The connections between neurons are represented by weights. These are adjusted during the training process to minimize the error between the predicted output and the actual target. We used ANN because they are able to process large amounts of data and better recognize patterns.

Approach 2: Through a process of feature extraction, ten segments are vertically extracted from each standardized MFCC. For each segment, we calculate the mean and standard deviation and add it to the development data frame as new features. Then they are fed to a Random Forest and K-Nearest Neighbor algorithm alongside other features.

Random Forest is an ensemble machine learning algorithm that combines multiple decision trees to make predictions [2]. It reduces overfitting and improves accuracy by using the average or majority of predictions made by the trees in the forest.

K-Nearest Neighbor (KNN) is a supervised machine learning algorithm used for classification. It works by finding the K data points in the training set that are closest to a new data point. It then uses their class or target values to make a prediction. [3]

C. Hyperparameters tuning

- Approach 1: Multi-layer fully connected ANN was used for sound classification with four hidden layers, and an output layer with seven nodes. The hyperparameter in ANN is the number of hidden layers. A simple grid search was performed, showing no significant improvement with an increased number of neurons. Therefore, the original values were kept.
- Approach 2: In this study, hyperparameter tuning was performed on two classifiers: K-Nearest Neighbors (KNN) and Random Forest. The best value of K in KNN was found to be 1. The best number of trees in Random Forest was found to be 90.
  Figure 6 implements a grid search (tested on 20 percent of the development data) and plots an accuracy curve for different values of K in the KNN algorithm. A similar analysis is also developed to comprehend the association between accuracy and the number of trees in the Random Forest (Figure 7).

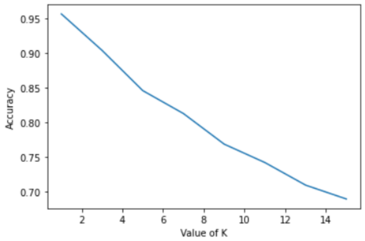

Fig. 6. Accuracy for Various numbers of K

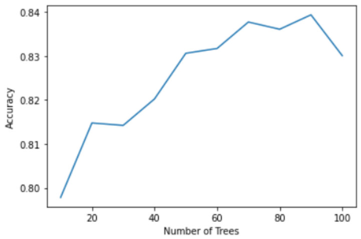

Fig. 7. Accuracy for Various numbers of tree

III. Results

The final accuracy of ANN model was 0.842. Number of neurons in each hidden layer were the hyperparameters, set to 1024, 512, 256, 128 and number of epochs equal to 100, based on prior research and experimentation. The use of k-nearest neighbors (k=1) resulted in an accuracy of 0.384, and using a random forest with 90 trees resulted in an accuracy of 0.412

The results demonstrate that the MFCCs trained in ANN algorithm achieved a higher accuracy compared to the second approach.

IV. Discussion

Consider the following aspects to enhance the performance of the result:

Convolutional Neural Networks (CNNs) can significantly enhance speech recognition projects by automatically extracting essential features from raw audio data [4]. These networks can perform end-to-end training, eliminating the need for manual feature engineering or alignment. This results in a more efficient and effective model that can handle raw audio data and produce accurate speech recognition outputs.

Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) networks, are a promising approach in speech recognition due to their capability to handle inputs of varying lengths. This feature makes LSTMs well suited for speech recognition tasks where the duration of speech inputs can differ [5]. Moreover, RNNs can be trained on extensive amounts of speech data, allowing them to learn intricate representations of speech features and enhance recognition accuracy. However, it's worth noting that other methods, such as Convolutional Neural Networks, may also have their own advantages in speech recognition and the choice of approach depends on the specific needs of the task.

V. References

[1] Oo, M. M. (2018). Comparative study of MFCC feature with different machine learning techniques in acoustic scene classification. International Journal of Research and Engineering, 5, 439-444.

[2] Cutler, A., Cutler, D. R., Stevens, J. R. (2012). Random forests. Ensemble machine learning: Methods and applications, 157-175.

[3] Jain, A. K., Dubes, R. C. (1988). Algorithms for clustering data. Prentice-Hall, Inc..

[4] Choi, K., Fazekas, G., Sandler, M., Cho, K. (2017, March). Convolutional recurrent neural networks for music classification. In 2017 IEEE International conference on acoustics, speech and signal processing (ICASSP) (pp. 2392-2396). IEEE.

[5] Weninger, F., Erdogan, H., Watanabe, S., Vincent, E., Le Roux, J., Hershey, J. R., Schuller, B. (2015). Speech enhancement with LSTM recurrent neural networks and its application to noise-robust ASR. In Latent Variable Analysis and Signal Separation: 12th International Conference, LVA/ICA 2015, Liberec, Czech Republic, August 25-28, 2015, Proceedings 12 (pp. 91-99). Springer International Publishing.