AI Project Progress Report

By:
Laura Wilson (lgw0020@auburn.edu)
Coordinator: Daniel Harrison(dah0052@auburn.edu)
Taylor Cross(tac0062@auburn.edu)

Dataset: https://huggingface.co/datasets/renumics/speech_commands_enriched
Title: Speech Command Classification Using Machine Learning Techniques

Introduction

The advancement in speech recognition technology has led to a growing interest in developing more sophisticated and responsive systems capable of understanding and executing spoken commands. This report details the progress we've made towards training a machine-learning model capable of classifying audio files into different speech commands. The primary application domain for this project includes robotics and large language models, where the system could interpret complex tasks through voice commands, focusing on keyword classification.

Statistics

The speech command dataset we are using has two versions of the dataset. We have decided to use version 0.01 which has 62, 727 audio files. In this version, there are 51093 training examples, 6799 validation examples, and 3081 testing examples. Thirty different words have been recorded in version 0.01: "Yes", "No", "Up", "Down", "Left" "Right" "On" "Off" "Stop" "Go" "Zero" "One" "Two" "Three" "Four" "Five", "Six", "Seven", "Eight", "Nine", "Bed", "Bird", "Cat", "Dog", "Happy", "House", "Marvin", "Sheila", "Tree", "Wow" each word associated with a value from 0 to 30 with 30 assigned to audio that has been labeled silence. The features that we visualize in our project are spectrogram, zero crossing rate, spectral centroid, spectral rolloff, Mel-Frequency Cepstral Coefficients or MFCC, and magphase. A spectrogram is a visual representation of the spectrum of frequencies of sound or other signals as they vary with time. Our MelSpectrogram transform is set to a sample rate of 16000 the default, n_fft, the size of FFT, equal 800, the length of hops between STFT window equal 16, and the number of mel filterbanks equal 23. Zero crossing rate is the rate of sign-change along a signal or the rate at which the signal changes from positive to negative or back. In our dataset the frame length is 400, hop length is 100. The spectral centroid indicates where the center of the sound is located. This is calculated as a weighted mean of frequencies present in the sound. If throughout the audio the frequencies are the same then the spectral centroid will be around the center and if the frequencies are high at the end of the sound then the centroid would be toward its end. Spectral rolloff is where the frequency is below a specified percentage of total spectral energy in our dataset this feature is set to 0.99 as well as 0.01 as a range of our audio. MFCC, Mel-Frequency Cepstral Coefficients, of a signal, are small sets of features, our mfcc is 13 with a sample rate of 16000, n_fft 400, hop length 160, and n_mels 23, which concisely describes the overall shape of a spectral envelope. Lastly, magphase splits the spectrogram into magnitude, and phase components.

Evaluation Metrics
Preliminary Results

In the development of speech recognition system, evaluating model performance comprehensively is crucial to ensure robust and accurate interpretation of spoken language. This paper discusses the application of a Multi-Layer Perceptron (MLP), a type of neural network known for its capability to model complex, non-linear relationships, in the context of speech recognition. To assess the MLP's efficacy, we have chosen F1-score, precision, and recall as our evaluation metrics, given their importance in classification tasks, including speech to text conversion where accuracy and reliability are paramount.

Evaluation Metrics Justification

- Precision in speech recognition is indicative of the system's ability to accurately transcribe spoken words without falsely interpreting background noise or irrelevant sounds as valid speech. High precision means that when the model predicts a specific word or phoneme, it is correct a high percentage of the time.
- Recall measures the system's capacity to correctly identify and transcribe all relevant speech sounds. A high recall value indicates that the model is effectively capturing the speech input without missing out on words or phonemes, which is critical for comprehensiveness in transcription.
- F1-Score offers a balanced measure of precision and recall, providing a single metric to gauge the overall performance of the speech recognition system. This is particularly useful in comparing the effectiveness of different models or configurations in a balanced manner, taking into account both false positives and false negatives.

Preliminary Results with MLP

Implementing an MLP for speech recognition, we processed a dataset comprising thousands of audio samples. The preliminary results were promising, with the MLP achieving an F1-score of 0.5626, precision of 0.5809, and recall of 0.5206. These metrics suggest a low level of accuracy in speech transcription, with the system being slightly more inclined towards capturing almost all relevant speech sounds at the expense of a minor increase in false positives.

Random Baseline Comparison

To contextualize these results, we established a random baseline model, designed to predict speech transcriptions based on the distribution of phonemes in the training set. As anticipated, the random model's performance was markedly inferior, with an F1-score, precision, and recall all hovering around 0.50. This just slightly better in contrast underscores the effectiveness of our MLP model over random guessing, highlighting its potential in speech recognition tasks.

Challenges and Moving Forward

The progress we have made has not been without challenges. There are some that we have overcome to make the progress that we have. Figuring out all the aspects of and interacting with the dataset was a major challenge in the beginning that kept us from being able to progress. This issue was solved by researching the dataset, interacting with the various components, and spending a lot of time. Visualizing the data so that we could better understand the features we wanted to extract was another challenge we had to face and put a lot of time into making it work. The main challenge for this was figuring out how to develop the code to properly display the feature that we wanted to see. In addition, each feature has its own set of functions that need to be used in the correct ways to properly graph them. There are other challenges that are persisting like figuring out how to extract the features that we want our model to learn from and actually getting to train the model, but we hope to resolve that in the near future as our next steps as we move forward in our project. After extracting our features, our plan is to define our model hyperparameter, construct the neural network, and start training. These tasks are by no means simple and due to the complexity of the dataset and computational requirements, there are sure to be more challenges to navigate in the future.