Speech Command Recognition using MFCC and Neural Networks

Submitted by:

Madhusudan

102116101

ABSTRACT

This project aims to classify spoken commands from a limited vocabulary using machine learning techniques. The Speech Commands dataset is preprocessed by extracting Mel-Frequency Cepstral Coefficients (MFCC) from audio signals. A neural network model is then trained to recognize commands such as "yes," "no," "up," and others. The system achieves a high degree of accuracy using a dense neural network architecture, employing dropout and regularization for improved generalization.

INTRODUCTION

Voice-controlled systems are becoming an integral part of modern technology, from smart home devices to personal assistants like Google Assistant and Alexa. This project focuses on classifying a set of pre-defined voice commands using speech recognition techniques. The goal is to build a model that can accurately predict which command was spoken from a limited set of commands. We utilize the Google Speech Commands dataset, a publicly available dataset that contains various spoken words.

DATASET DESCRIPTION

- **Dataset Source:** Google Speech Commands Dataset v0.02
- **Number of Classes:** 10 (Yes, No, Up, Down, Left, Right, On, Off, Stop, Go)
- **Dataset Structure:** The dataset consists of one-second audio clips sampled at 16 kHz.
- **Preprocessing:** Each audio clip is converted to its Mel-Frequency Cepstral Coefficients (MFCC), which capture the most essential audio features for speech recognition. Thirteen MFCC features are extracted per audio file.

PREPROCESSING

Each audio clip is loaded and resampled to a standard sampling rate of 16 kHz to ensure consistency. We then extract MFCC features, which are known to perform well for speech and audio signal analysis. These features are stored as a matrix representing the spectral properties of each audio file. The mean of each MFCC feature vector is used to form the final feature set.

MODEL ARCHITECTURE

Input Layer:

- Takes the shape of the feature set (`input_shape=(X_train.shape[1],)`), which corresponds to the number of MFCC coefficients.

Hidden Layers:

- **Layer 1:** 256 neurons, ReLU activation, followed by 30% dropout.
- **Layer 2:** 128 neurons, ReLU activation, followed by 30% dropout.
- **Layer 3:** 64 neurons, ReLU activation, followed by 30% dropout.

Output Layer:

- Uses a softmax activation function to classify into `n_classes` categories (10 commands), where each unit in the output layer corresponds to a specific command.

TRAINING PROCESS

Training Set Size: 80% of the total dataset
Validation Set Size: 20% of the total dataset
Batch Size: 32
Epochs: 15
Hyperparameters:

- Optimizer: Adam
- Loss Function: Sparse Categorical Crossentropy The model is trained for 15 epochs with a batch size of 32, utilizing 20% of the training data for validation. Early stopping is used to prevent overfitting, stopping the training when the validation loss starts to increase.

RESULTS

Training Accuracy: Achieved an accuracy of 95% on the training set after 15 epochs.

- **Validation Accuracy:** Achieved 93% validation accuracy.
- **Training and Validation Loss:** The model's training and validation loss steadily decreased over time, with minimal overfitting observed due to the dropout layers and regularization techniques used.

Accuracy and Loss Curves:

- The accuracy and loss curves for both training and validation are shown below. The model converged well, achieving high accuracy while keeping the validation loss low.

REFRENCES

- Warden, P. (2018). "Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition". arXiv:1804.03209
- Librosa: Python library for audio and music analysis. Librosa
- TensorFlow: Deep Learning framework. TensorFlow

ASSETS:

DATASET:

https://drive.google.com/drive/folders/1zqQct8AvoCZcXBeZPz7ErNv5KGMQpU-p?usp=share_link