Report on Speech Command Recognition Project Using TDNN

MID SEMESTER LAB EVALUATION

CONVERSATIONAL AI: SPEECH PROCESSING AND SYNTHESIS(UCS749)

Submitted by:

kavay khurana

102103645

4CO23

BE Fourth Year, COE

Submitted to:

Dr. Raghav B.Venkataramaiyer

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Computer Science and Engineering Department

Thapar Institute of Engineering and Technology, Patiala

September 2024

Summary of Research Paper Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition:

Pete Warden from Google Brain has introduced a new dataset designed to advance the development of keyword spotting systems. The dataset features a diverse collection of common words and commands, recorded in a variety of accents and real-world environments. Its goal is to train speaker-independent models that can operate efficiently on devices.

To ensure the dataset's accessibility and usefulness, Warden has released it under a Creative Commons license. The data includes one-second utterances of single words, accompanied by background noise samples to enhance model training. This approach aims to simulate the challenges faced by keyword spotting systems in everyday settings.

Warden highlights the importance of open datasets in fostering collaboration and innovation within the research community. By sharing this dataset, he hopes to encourage the development of more robust and accurate keyword spotting technologies.

1. Introduction

In this project, the primary goal was to build a robust speech recognition model using Time-Delay Neural Networks (TDNN). The dataset consisted of audio files stored in subfolders, where each subfolder represented a distinct category, and the files were in .wav format. Given TDNN's efficiency in handling temporal dependencies, it was chosen as the core neural network architecture for this task. The project involved several key stages, including data preprocessing, feature extraction, model building, fine-tuning, and data augmentation.

2. Dataset Overview

The dataset was structured into multiple categories, each containing several .wav audio files. These audio clips varied in duration and content, representing different voice patterns. To optimize the model for better generalization, a consistent number of samples per category was required, which was later addressed by generating synthetic data through augmentation.

3. Data Preprocessing

Before feeding the audio data into the TDNN model, several preprocessing steps were performed:

- **Trimming Silence**: Any leading or trailing silence in the audio files was removed to ensure that the model only trained on meaningful sound data.
- **Normalization**: Each audio file was normalized to have a uniform loudness, preventing discrepancies due to variations in volume across different samples.

4. Feature Extraction using MFCC

For feature extraction, **Mel-Frequency Cepstral Coefficients (MFCC)** were computed from each audio file. MFCC is a widely used technique in speech processing because it effectively captures the essential features of human voice signals:

- **13 MFCC features** were extracted per audio file, which served as the input data for the TDNN model. This conversion from raw audio to a numerical feature vector allowed the neural network to learn patterns more efficiently.

5. TDNN Model Architecture

A Time-Delay Neural Network (TDNN) was implemented to process the MFCC features. The structure of the TDNN included multiple layers, specifically designed to capture time-related dependencies in the audio data:

- **Multiple Convolutional Layers (Conv1D)**: Used to extract temporal features from the input MFCCs.
- **Batch Normalization & Dropout**: Added to improve the model's generalization ability and prevent overfitting.
- **Fully Connected Layers**: Two dense layers were added to further refine the learned features before the final classification.

After training the model on the preprocessed dataset, the **training accuracy reached 89%**, and the **testing accuracy peaked at 97.90%**.

6. Model Fine-Tuning on New Dataset

After obtaining satisfactory results with the initial dataset, the next objective was to fine-tune the model on a smaller dataset of **30 samples per category**. To accomplish this:

- **Freezing Layers**: All layers of the pre-trained TDNN model were frozen except for the last 4 layers, allowing these layers to be updated based on the new data.
- **Adding New Layers**: Additional Conv1D layers were introduced to help the model capture new voice patterns specific to this smaller dataset.

However, this fine-tuning resulted in poor performance, yielding a **training accuracy of only 9%**. This suggested that the new dataset was too small or lacked sufficient variability to enable the model to generalize effectively.

7. Synthetic Data Generation through Data Augmentation

To address the poor performance during fine-tuning, data augmentation techniques were applied to increase the size and variability of the dataset. The original dataset was augmented from **1,500 rows to around 19,000 rows** using the following techniques:

- **Adding Noise**: Random white noise was added to the audio samples.
- **Speed Manipulation**: The speed of the audio was altered (both increased and decreased) to introduce variability.
- **Pitch Shifting**: The pitch of the audio was modified, simulating different vocal tones.
- **Time Shifting**: The audio samples were shifted forward or backward in time.
- **Volume Adjustment**: The audio's volume was altered to simulate different recording conditions.
- **Reversing Audio**: Some audio files were reversed to introduce further variation.

This data augmentation process significantly increased the dataset size and diversity, which proved crucial for improving the model's performance during fine-tuning.

8. Synthetic Data Generation through Data Augmentation

After augmenting the data, the fine-tuned TDNN model was re-trained on the new dataset. This led to a remarkable improvement in model performance:

- Training Accuracy: 99%
- Validation Accuracy: 91%

The increase in both the size and variability of the dataset through augmentation allowed the model to generalize better, resulting in a significantly higher accuracy on the validation set compared to the earlier attempts with a smaller dataset.

9. Conclusion

In this project, the TDNN model demonstrated strong performance in recognizing speech patterns when trained on a sufficient and diverse dataset. While the initial fine-tuning attempt on a small dataset was unsuccessful, data augmentation techniques allowed for the creation of a large synthetic dataset, leading to a substantial increase in model accuracy. The project underscores the importance of both feature extraction (MFCC) and data augmentation in speech recognition tasks and highlights TDNN’s effectiveness in capturing temporal dependencies in audio data.

Future work may involve exploring more advanced augmentation techniques, fine-tuning hyperparameters, or testing other architectures such as LSTM or CNN to further enhance the model's performance on speech recognition tasks.