Final-Project
Detailed Latex Report

Fuzayil Bin Afzal Mir

17/02/2021

1 Speech Command Recognition

• Theory

Speech is one of the most effective way for human and machine to interact. This project aims to build Speech Command Recognition System that is capable of predicting the predefined speech commands.

Speech recognition, also known as automatic speech recognition (ASR), computer speech recognition, or speech-to-text, is a capability which enables a program to process human speech into a written format. While it's commonly confused with voice recognition, speech recognition focuses on the translation of speech from a verbal format to a text one whereas voice recognition just seeks to identify an individual user's voice.

• Data Set

A data set is a collection of data.Every data set can be created in a different way.Some of the data sets can be machine generated data sets and some may be data that's recorded from human observations.

The data set of the given speech recognition system consists of the following pre recorded commands:

1)Forward

2)Back

3)Left

4)Right

5)Stop

The sampling rate of the recorded voice commands is 16 KHz and 80 utterances of each command have been generated.All the samples have been trimmed to 1 second.

About the code:

The code has been written using the python programming language and the packages and libraries used are : numpy, librosa, random, tensorflow and soundfile etc.

1.1 Kapre:

It is used for audio processing in keras layers.

1.2 Sound File:

It can read or write sound files.

1.3 ffmpeg:

It is an audio line tool that converts audio or video formats.It can also capture and encode in real time from various hardware and software sources such as TV capture card.

1.4 Import OS:

It provides the functions for interacting with the operating system.

1.5 Import Numpy:

It is used for working with arrays.It has functions for working in domain of linear algebra,fourier transform and matrices.

1.6 Import Librosa:

It is used for music and audio analysis eg. python module for audio and music processing.

1.7 Import Random:

This module is used to generate random numbers.

1.8 Import Tensorflow:

Tensorflow is an open source library.It is used to create large scale neural network with many layers.
It is mainly sed for deep learning or deep learning projects such as classification, perception, understanding, discovering etc.

1.9 Keras:

It is an powerful and easy to use free open source python library for developing and evaluating deep learning models.

1.10 tensorflow.keras:

Along with tensorflow keras allows to define and train neural network models in just a few lines of code.

1.11 Import sklearn:

It is a free software mahine learning library for the python programming language.

1.12

1)Now we mount the google drive so that the data set present in drive can be used in the given program.
2) Then we need to run the generate data portion in the code and seperate the data into the form of arrays.

1.13 Splitting data into Train and Test:

In this section we take the data and divide it into the two subsets.

Subset 1: Train Data set: It is used the train the machine learning model.

Subset 2: Test Data set: It is used to evaluate the machine learning model.

1.14

1) After this we have to extract the features by using mfcc (mel frequency cepstral coefficients).
2) Then the data is to be loaded And the model is trained by running the train model portion of the code.

1.15 Check Attention:

In this part we will get 2 type of graphs one represents the original recorded data using the audacity software and the another graph represents the analog signal.

Note: The analog signal will be represented using the thin line graph.

1.16 Testing the Model:

Here we can check the acuuracy of our model.After running this part we get a chance to record an audio and then we will get an output from our model and we can check whether that output matches our input or not.

Instructions: 1)The Code is written using Google Colab.

2)Open ColabNotebook.ipynb and write the code in different consecutive cells.

3)Upload the Data Set to Colab.

4)Run the cells in the same order.

5)Locate the folder where you save your model.h5 file.

6)Start speaking when you see mike in the bottom right pane of the task bar or see red blinking dot in the title bar.

7)The repeated errors cab be minimised by setting the mike recorder time in the previous record function up to 3 sec.

• Description

1)Using Convolutional layers ahead of LSTM is shown to improve performance in several research papers.
2)BatchNormalization layers are added to improve convergence rate.
3)Using Bidirectional LSTM is optimal when complete input is available. But this increases the runtime two-fold.
4)Final output sequence of LSTM layer is used to calculate importance of units in LSTM using a FC layer.
5)Then take the dot product of unit importance and output sequences of LSTM to get Attention scores of each time step.
6)Take the dot product of Attention scores and the output sequences of LSTM to get attention vector.
7)Add an additional FC Layer and then to output Layer with SoftMax Activation.
8)The model is successfully built and has achieved the highest accuracy of 98.626%.

The End