SPEECH COMMAND MODEL

Muneeb Ahmad Sheikh

February 7, 2021

- **Language Used:**

  PYTHON

- **Libraries and Packages Used:**

  KAPRE, SCIKIT LEARN, SOUND FILE, TENSORFLOW.

- **OVERVIEW:**

  Speech Command Recognition is changing voices in to text form,Speech command recognition is present in a wide range of devices and utilized by many HCI interfaces. In many situations, it is desirable to obtain lightweight and high accuracy models that can run locally.
  Some speech recognition systems require "training" (also called "enrollment") where an individual speaker reads text or isolated vocabulary into the system. The system analyzes the person's specific voice and uses it to fine-tune the recognition of that person's speech, resulting in increased accuracy. Systems that do not use training are called "speaker independent"[1] systems. Systems that use training are called "speaker dependent".
  Speech recognition applications include voice user interfaces such as voice dialing (e.g. "call home"), call routing (e.g. "I would like to make a collect call"), domotic appliance control, search key words (e.g. find a podcast where particular words were spoken), simple data entry (e.g., entering a credit card number), preparation of structured documents (e.g. a radiology report), determining speaker characteristics,[2] speech-to-text processing (e.g., word processors or emails), and aircraft (usually termed direct voice input).

- **DATA:**

1: Set 16KHz as sampling rate.

2: Record 80 utterances of each command.

3: Save samples of each command in different folders.

  - Dataset/forward.
  - Dataset/back.
  - Dataset/left.
  - Dataset/right.
  - Dataset/stop.

- **RUN:**

The Code is written using Google Colab:

1: Open ColabNotebook.ipynb and change Runtime to GPU.

2: Upload Speech-Recognition-Project/Data to Colab.

3: Change data-dir in all cells to point to Speech-Recognition-Project/Data.

4: Run the cells in the same order in Notebook Test.

- **TEST:**

1: Locate the folder where you save your model.h5 file.

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

- **MODEL SUMMARY:**

Model: "Attention"

Layer (type)	Output Shape	Param #	Connected to
Input (InputLayer)	[(None, 49, 39, 1)]	0	
Conv1 (Conv2D)	(None, 49, 39, 10)	60	Input[0][0]
BN1 (BatchNormalization)	(None, 49, 39, 10)	40	Conv1[0][0]
Conv2 (Conv2D)	(None, 49, 39, 1)	51	BN1[0][0]
BN2 (BatchNormalization)	(None, 49, 39, 1)	4	Conv2[0][0]
Squeeze (Reshape)	(None, 49, 39)	0	BN2[0][0]
LSTM_Sequences (LSTM)	(None, 49, 64)	26624	Squeeze[0][0]
FinalSequence (Lambda)	(None, 64)	0	LSTM_Sequences[0][0]
UnitImportance (Dense)	(None, 64)	4160	FinalSequence[0][0]
AttentionScores (Dot)	(None, 49)	0	UnitImportance[0][0] LSTM_Sequences[0][0]
AttentionSoftmax (Softmax)	(None, 49)	0	AttentionScores[0][0]
AttentionVector (Dot)	(None, 64)	0	AttentionSoftmax[0][0] LSTM_Sequences[0][0]
FC (Dense)	(None, 32)	2080	AttentionVector[0][0]
Output (Dense)	(None, 5)	165	FC[0][0]

- The model is accomplished with an maximum accuracy of 99.8%