Report

February 5, 2021

- **OVERVIEW:**

  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.It focuses on the translation of speech from a verbal format to a text one.It's working applications is Google Home Assistant that can place space trivia with you and make financial transactions when requested. In similar way Speech voice recognition model is based on concepts of Convolution, LSTM , Attention and recognise pretrained voice with accuracy of **99.9%**.

- **DATA:**

  1: Set 16KHz as sampling rate.
  2: Record 80 utterances of each command.
  3: Save samples of each command in different folders.

* Data/forward.
* Data/back.
* Data/left.
* Data/right.
* Data/stop.

- **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.

- **The model is successfully built and has achieved the highest accuracy of 99.9%**

- **Model Summary**

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]

- **RUN:**

  The Code is written using Google Colab:

1. Open ColabNotebook.ipynb and change Runtime to GPU.
2. Upload Speech-Recognition/Speech to Colab.
3. Change data-dir in all cells to point to Speech-Recognition/speech.
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.

- **Language Used:**

  PYTHON

- **Libraries and Packages Used:**

  KAPRE, SCIKIT LEARN, SOUND FILE, TENSORFLOW.