Slurred Speech Recognition using Deep learning and Transfer learning

Report submitted in fulfillment of the requirements
for the Exploratory Project of

Fourth Year B.Tech.

by

Kumar Shivam Ranjan
Madhav Bansal

Under the guidance of

Dr. Hari Prabhat Gupta

Department of Computer Science and Engineering
INDIAN INSTITUTE OF TECHNOLOGY (BHU) VARANASI
Varanasi 221005, India
November 2021

Dedicated to

My parents, teachers,.....

Declaration

I certify that

1. The work contained in this report is original and has been done by myself and the general supervision of my supervisor.
2. The work has not been submitted for any project.
3. Whenever I have used materials (data, theoretical analysis, results) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references.
4. Whenever I have quoted written materials from other sources, I have put them under quotation marks and given due credit to the sources by citing them and giving required details in the references.

Place: IIT (BHU) Varanasi
Date: **20/11/2021**

Kumar Shivam Ranjan , Madhav Bansal
B.Tech. or IDD Student
Department of Computer Science and Engineering,
Indian Institute of Technology (BHU) Varanasi,
Varanasi, INDIA 221005.

Certificate

*This is to certify that the work contained in this report entitled "**Slurred Speech Recognition using transfer learning** " being submitted by **Kumar Shivam Ranjan(18075031) and Madhav Bansal(18075036)** carried out in the Department of Computer Science and Engineering, Indian Institute of Technology (BHU) Varanasi, is a bona fide work of our supervision.*

Place: IIT (BHU) Varanasi
Date: **20/11/2021**

Dr. hari Prabhat Gupta
Department of Computer Science and Engineering,
Indian Institute of Technology (BHU) Varanasi,
Varanasi, INDIA 221005.

Acknowledgments

I would like to express our sincere gratitude to our Teaching Assistant **Mr. Atul Chaudhary** and our supervisor **Dr.Hari Prabhat Gupta** who helped us whenever we needed them and our parents for their belief and support.

Place: IIT (BHU) Varanasi
Date: **20/11/2021** **Kumar Shivam Ranjan , Madhav Bansal**

Abstract

Deep learning has evolved a lot to develop end to end state of the art speech Recognition System. End to end deep learning model for automatic speech Recognition is much simpler than traditional speech systems which tend to perform poorly in noisy enviroments. Deep learning End to End models doesn't need hand designed components to model background noise and speaker variation. Moreover, the concept of phoneme is not that relevant in end to end speech recognition systems unlike traditional speech recognition systems.In recent times, bidirectional GRU based deep learning model or RNN-CTC based models have shown the results which has provided a new benchmark for end to end models for automatic speech recognition.
The well known Automatic Speech Recognition Systems are Alexa, Siri , Google Home etc. This class of applications starts with clip of spoken audio clip in a particular language and extracts the words that were spoken as text. There are also called Speech to Text Algorithms.

Contents

Chapter 1

Introduction

1.1 Overview

The conventional speech recognition system heavily depends on multiple pipelines consisting of several algorithms with hand-engineered processing stages. They tend to perfrom poorly with noisy environments and speaker variation. The introduction of end to end deep learning models have not onlty significantly improved the performance of speech recognition systems but also have decreased the training time and reduced the efforts for heavily engineered processing pipelines.
In this report we present our end to end deep learning model implementation that not only converts an audio signal into human spoken transcripts but also takes into account a special usecase for speech recognition.

1.2 Problem Statement

The state of the art Automatic Speech Recognition(ASR) can greatly improve the lives of those with speech impairements. However , the end to end deep learning Automatic Speech Recognition System trained from 'normal' speech tend to perform poorly for those who have speech impairements either due to accident or disease. The purpose of this project is to improve ASR for people who have speech impairements. In other words, we have to develop an ASR model that works on personalised non-standard speech.

1.3 High level Solution overview

The Automatic Speech Recognition system such as Amazon Alexa, apple Siri, Google Home etc, starts with a clip of spoken audio and extracts the words that were spoken in that audio as text irrespective of the speaker variation. There are also called Speech to Text Algorithms.
For Speech-to-Text problems, our training data consists of:

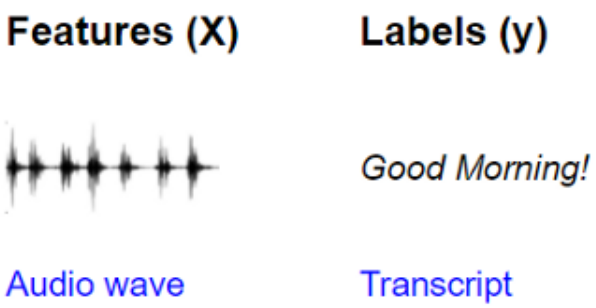

Figure 1.1 ASR uses audio waves as input and text transcript as target

However, the current state-of-the-art ASR models can yield high word error rates and character error rates for speakers with speech impairments either due to accidents or disease.So, we cannot use the current end to end ASR models for such people. This is the reason that **Personalised ASR for atypical speech with limited data** is the need of the hour. We start with a high-quality ASR model trained on thousands of hours of standard speech and then we fine-tune parts of the model to an individual with non-standard speech. Tranfer learning is used for fine tuning the base model.

The major issue with slurred speech recognition was the difficulty in collecting enough data to train a state-of-the-art recognizer for individuals. ASR models are trained on hundreds of hours of speech dataset. Acquiring much dataset from a single volunteer is almost impossible.We overcome this issue by first training a base ASR end to end deep learning model on a large corpus of typical speech, and then training a personalized model using a much smaller dataset with the targeted non-standard speech characteristics.

Chapter 2

Theory

2.1 Data pre-processing workflow

The very first phase for this project is to obtain high accuracy ASR end-to-end deep learning model. We will start by implementing the same. We'll be training on a subset of LibriSpeech Dataset, comprising 100 hours of transcribed audio data from different speakers.
For this problem, our training data consists of:
Input features (X): audio signals of human spoken words.
Target labels (Y): a text transcript of what was spoken in the audio.
The goal of the model is to learn how to take the input audio and predict the text content of the words and sentences that were spoken.

Following is the workflow for data preprocessing:

- Load audio files
- Resampling
- Raw audio data augmentation
- Mel Spectograms generation
- Conversion to MFCC
- SpecAugmentation (Spectogram Data augmentation)

2.2 Model architecture

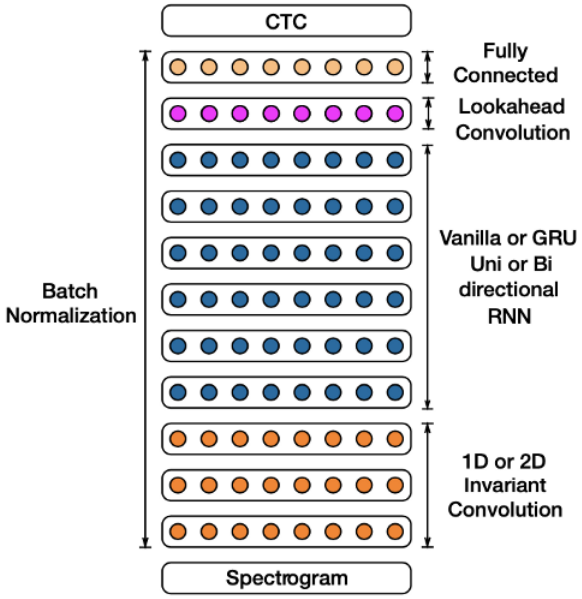

Figure 2.1 ASR Model Architecture

The core of this architecture is the Convolutional neural network(CNN) and Recurrent neural network(RNN) precisely gated recurrent unit trained to ingest speech spectograms and generate text transcripts.
Suppose our training set consists of :

$$X = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), ...\}$$

where x is the single utterance and y is the text transcript.
Each data point $x^{(i)}$ is of length $T^{(i)}$ where every time-step is a vector of amplitudes.So i$^{\text{th}}$ data point can be represented as ${x^{(i)}}_t$ where t= 1, 2, .. $T^{(i)}$. Our audio features is the spectrograms that we got as data preprocessing result. The goal of our model is to update the weights of the CNN and RNN layers so as to be able to convert the input array x into character probabilities per time-step.

2.3 Model Workflow

Our Model is composed of CNN layers as well as RNN layers packed with Connectionist temporal classification loss algorithm to demaracte each alphabets or characters of spoken words in the audio file.
A high level model understanding is made by:

- CNN layers composed of few Residual CNNs that take mel-spectograms as input and produce feature maps of the same.
- After CNN, we have few RNN(ranging from 5 to 7) layers , precisely some bidirectional GRU(gated recurrent unit) that takes those feature maps as produced by CNN layers as a consecutive sequence of timesteps or frames. The feature maps which are continous representation of the audio waves are eventually converted into discrete representation.
- A linear layer is at the end that take output from the RNNs and produce character probabilities for each timestep or frame.

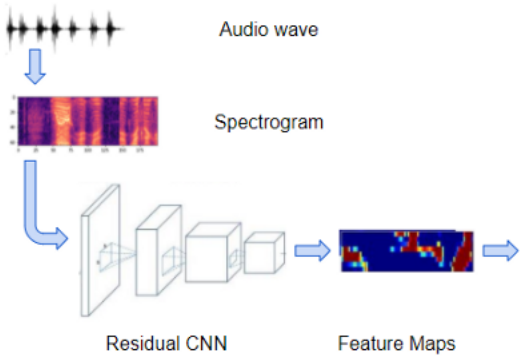

Figure 2.2 Spectrogram as input to CNN

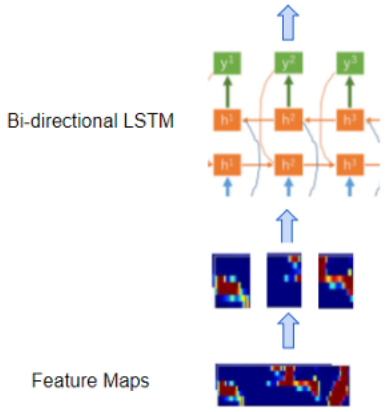

Figure 2.3 Feature maps as input to RNN

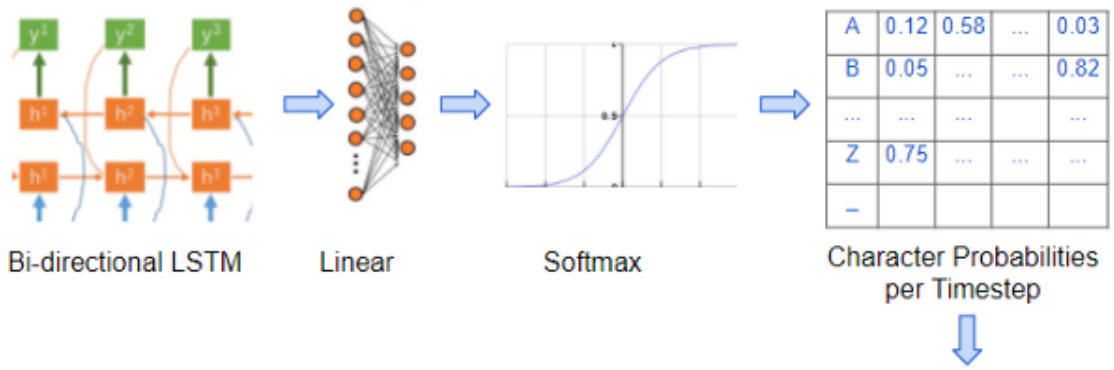

Figure 2.4 Character probabilities for each frame

2.4 Model Evaluation metric

Model Evaluation is very important phase of machine learning or deep learning pipeline.Once, we train our model, we must evaluate its performance. The metric used in ASR systems is called Word error rate(WER) and character error rate(CER). They are used to compare the transcript predicted by our model to actual transcript and generates a number that represents how much difference there is in the predicted and the actual transcript.
There are three terms related to word error rate:

- Deletion: count of words that are present in the transcript but not in prediction
- Insertion:count of words that are present in the transcript but not in prediction
- Substitution: count of words that are altered between prediction and transcript

$$WER = \frac{Deletion + Insertion + Substitution}{total}$$

2.5 Connectionist Temporal classification

Connectionist Temporal classification is used to establish demarcation or boundaries between spoken characters. Since our input is continous sound waves and the output is discrete, there is a need to align our input to output because there is no clear boundaries between elements. It removes the need to manually provide the aligment as a part of labelled training set.
CTC algorithm takes the output of the RNN (character probabilities) and generate some valid character sequence. There is concept of "blank" token introduced in CTC represented by "-" to handle alignments and multiple repeated characters in the spoken audio. Character probabilities also include the the blank token probability.

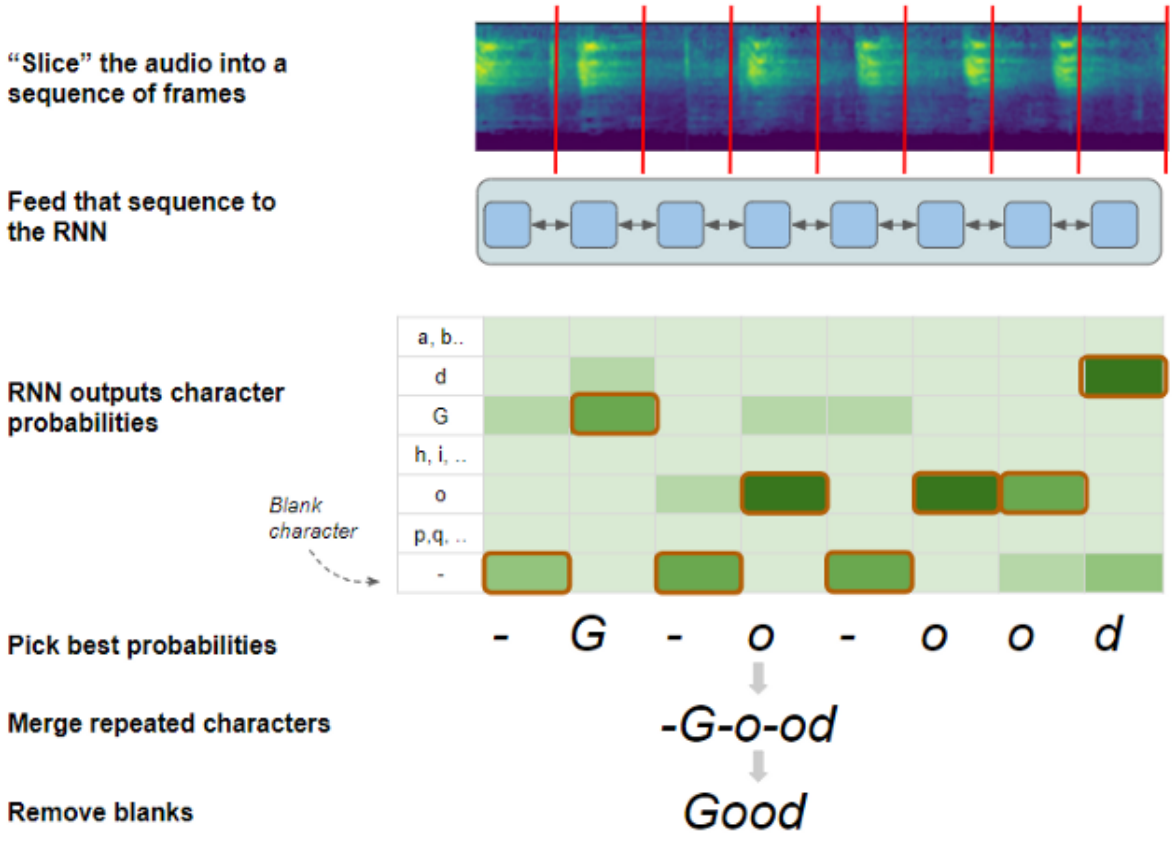

Figure 2.5 CTC Decoding

2.6 CTC Loss

CTC loss is calculated as probability of our model predicting the correct transcript. This algorithm first generates all valid character sequences. The algorithm follows the following paradigm:

- It keeps track of only those characters that occur in the target and removes the rest
- It selects only those alignments which follows the same order as the target transcript
- It computes the character probabilities for each time-step which in turn helps in calculating the probability of generating all valid character sequences.
- It is computed as negative logarithm probability of all possible valid character sequences.

If we have 3 possible alignments , CTC will be calculating

$$P(y, alignment1|X), P(y, alignment2|X), P(y, alignment3|X)$$

where y is the label and X is the split spectrogram. After that, we would like to eliminate the alignment information to get only P(y |X).

$$loss = -log(P(y|X))$$

- when P(y |X) aproaches 0, loss aproaches infinity
- when P(y |X) aproaches 1, loss aproaches 0
- We are training the model to learn and maximise P(y |X). When P(y |X) aproaches 1 , loss aproaches 0 , then we will not update weights of the model.

2.7 Transfer Learning

Chapter 3

Project Work

3.1 Libraries/APIs used

3.2 Results

Chapter 4

Conclusions and Discussion

4.1 Conclusion

4.2 Limitations

4.3 Future Scope

Bibliography