Run-Time Conversion of Speech from One Language to Another using Deep Learning

Progress Report

In fulfillment of the requirements for the

NU 302 R&D Project

At NIIT University

Submitted by

Deeptonabho Dutta
Gaurav Mundhra
K.N Raviteja
Sabyasachi Mishra
Shubhangi Gupta

Area - CSE

NIIT University

Neemrana

Rajasthan

CERTIFICATE

*This is to certify that the present research work entitled "**Run-time conversion of speech from one language to another using deep learning**" being submitted to NIIT University, Neemrana, Rajasthan, in the fulfillment of the requirements for the course at NIIT University, Neemrana, embodies authentic and faithful record of original research carried out by Deeptonabho Dutta, Gaurav Mundhra, K.N Raviteja, Sabyasachi Mishra, Shubhangi Gupta, student/s of B Tech (CSE) at NIIT University, Neemrana,. She /He has worked under our supervision and that the matter embodied in this project work has not been submitted, in part or full, as a project report for any course of NIIT University, Neemrana or any other university.*

Prof. Prosenjit Gupta,
Professor(CSE),
NIIT University

Mr. Vikas Kumar Malviya,
Assistant Professor (CSE),
NIIT University

LIST OF FIGURES

LIST OF TABLES

TABLES **PAGE NO.**

TABLE OF CONTENTS

RATIONALE OF THE WORK

Language is a genuine problem when difficult to understand. Through our research, we aim to considerably solve the following problems:

- **Lingual Barrier -** To make language easy to understand and also to widen our knowledge in the fields of machine learning and deep learning.
- **Conferences** - To make speeches in international conferences accessible to all the dignitaries.
- **Education** - To instantaneously translate a teacher's lecture to different languages in a classroom.
- **Tourism** – To enable a tourist to communicate effectively with people in a new country.
- **Blind People** - To translate on-screen text to a blind person's native language.

A number of models already existed in the realm of speech translation although each had its own drawback.

Pre-1970

1. Speech recognition was implemented for less vocabulary of words(16-200 words).
2. User had to give a pause after every word for the system to recognize.

1970-2000

1. Introduction of HMM model.
2. 30 seconds of speech required 100 minutes to be translated into another language.

2000s

1. Introduction of LSTM which outperformed HMM models.
2. There was a increase in the performance and decrease in word error.

LITERATURE REVIEW

We read a number of research papers, focusing on speech recognition and speech translation, and tried to analyze the merits and demerits of the methods proposed in each of them.

1. Deng, Fellow and Xiao Li**, Machine Learning Paradigms for Speech Recognition: An Overview**

   This paper provides an overview of modern ML techniques as utilized in the current and as relevant to future ASR(Automatic Speech Recognition) research and systems. The article is organized according to the major ML paradigms that are either popular already or have potential for making significant contributions to ASR technology. The concepts elaborated in this overview include: generative and discriminative learning; supervised, unsupervised, semi-supervised, and active learning; adaptive and multi-task learning; and Bayesian learning.

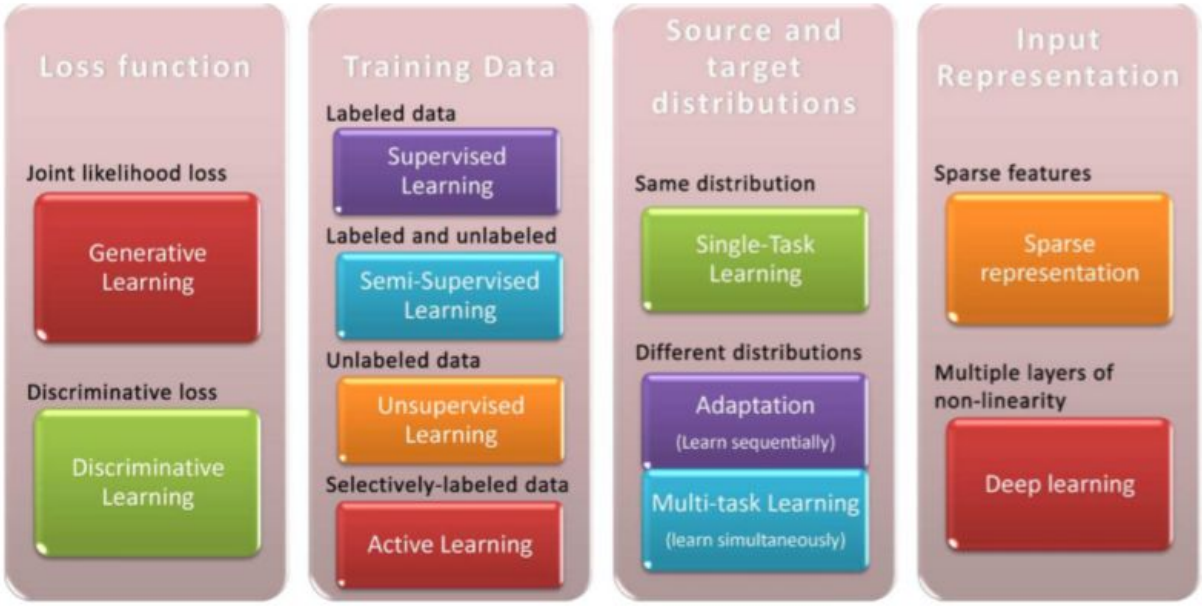

Fig 6.1

2. Janna Escur i Gelabert , Xavier Gir´o-i-Nieto, Marta Ruiz Costa-Juss`a, **Exploring Automatic Speech Recognition with TensorFlow**

   This bachelor's thesis focuses on using deep learning techniques to build an end-to-end Speech Recognition system. It overviews the most relevant methods carried out over the last several years. Then, it lays out one of the latest proposals for this end-to-end approach that uses a sequence to sequence model with attention-based mechanisms. Next, it successfully reproduces the model and test it over the TIMIT database

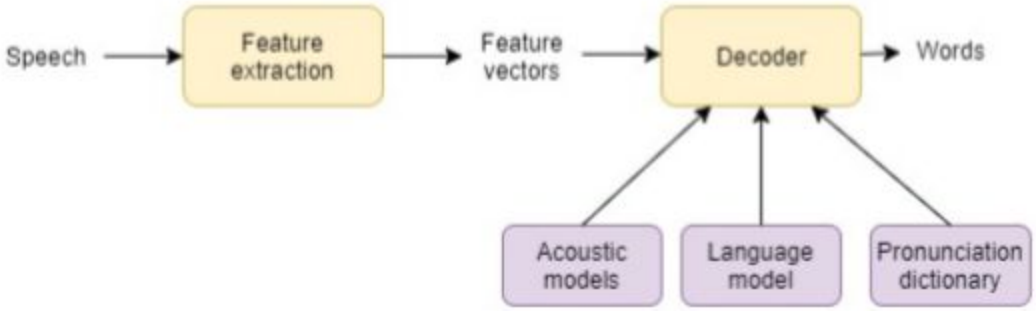

Fig 6.2

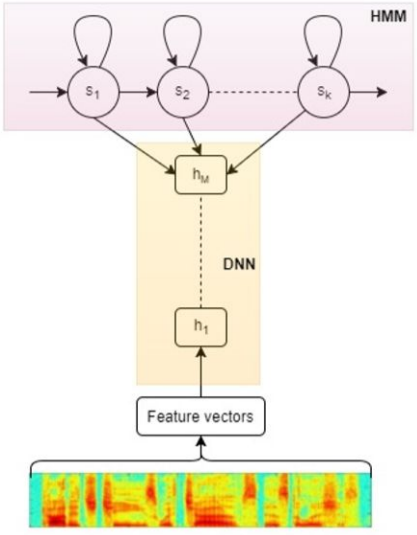

Fig 6.3

3. Wouter Gevaert, Georgi Tsenov, Valeri Mladenov, **Neural Networks used for Speech Recognition**

   This paper presents an investigation of speech recognition classification performance, performed using two standard neural networks structures as the classifier. The utilized standard neural network types include Feed-forward Neural Network (NN) with back propagation algorithm and a Radial Basis Functions Neural Networks.

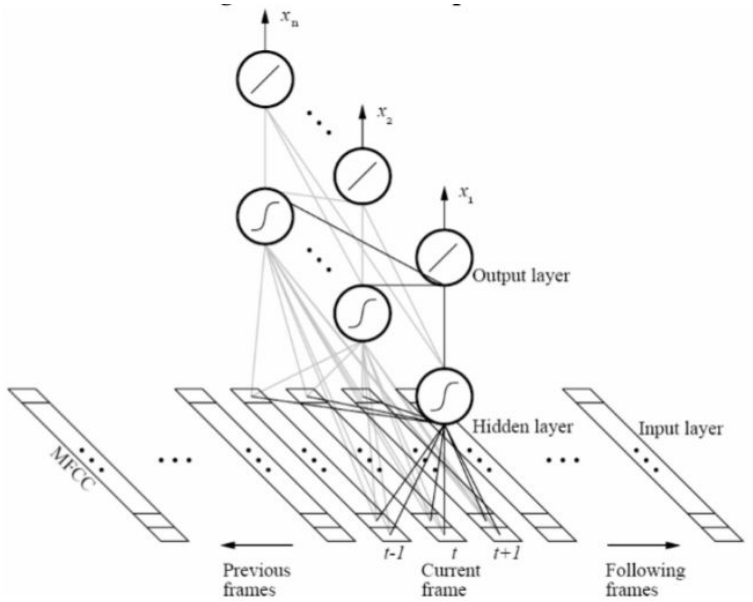

Fig 6.4

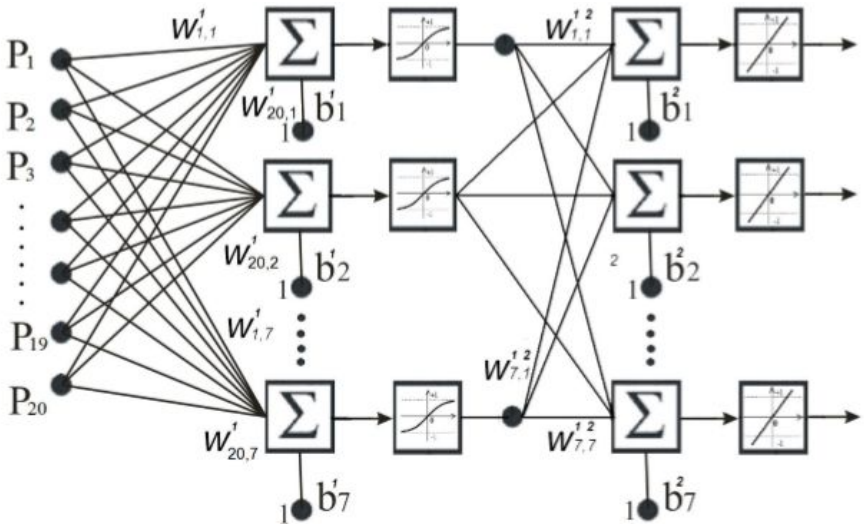

Fig 6.5

4. Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, **Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation**

   Neural Machine Translation (NMT) is an end-to-end learning approach for automated translation, capable of overcoming many of the weaknesses of conventional phrase-based translation systems. Unfortunately,they are computationally expensive both in training and in translation inference and lack robustness,particularly when input sentences contain rare words. Google's Neural Machine Translation system attempts to address many of these issues. Their model consists of a deep LSTM network using residual connections as well as attention connections from the decoder network to the encoder. This method provides a good balance between the flexibility of "character"-delimited models and the efficiency of "word"-delimited models, naturally handles translation of rare words, and ultimately improves the overall accuracy of the system.

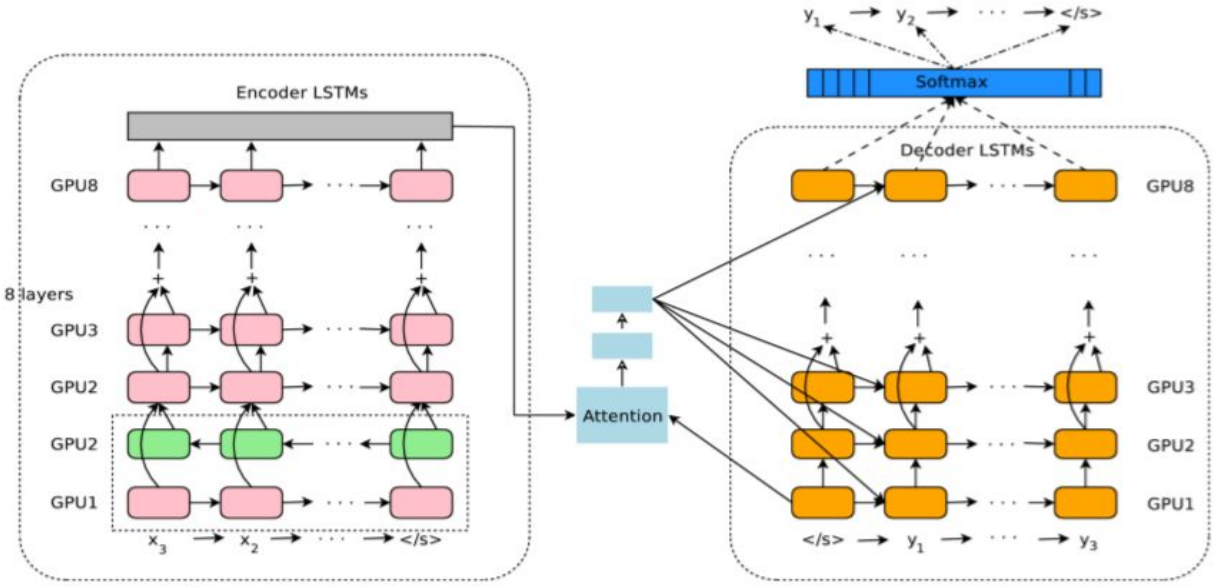

Fig 6.6

5. Hervé Blanchon, Laurent Besacier ,
COMPARING STATISTICAL MACHINE TRANSLATION (SMT) AND NEURAL MACHINE TRANSLATION (NMT) PERFORMANCES

This thesis compares the performances of the two standard translation methods - SMT and NMT.

	SMT	NMT
Core element	Words	Vectors
Knowledge	Phrase table	Learned weights
Training	Slow Complex pipeline	Slower More elegant pipeline
Model size	Large	Smaller
Interpretability	Medium	Very low Opaque translation process
Introducing ling. knowledge	Doable	Doable (yet to be done!)
Open source toolkit	Yes (Moses)	Yes (many!)
Industrial deployment	Yes	Yes (now at google, systran, wipo)

Fig 6.7

OBJECTIVES

Building a real time translation system between any language. Initially, we divided our project into three phases:

- To build a neural network which recognizes speech.
- To build a translation model which converts text in one language to another.
- To perform a speech synthesis on the translated text and convert it into an audio.

However, later we narrowed our scope to

- Recognition (using Machine Learning and Deep Learning), and
- Translation, performed through a model which recognises numbers spoken in English (eg.“one”) and converts it into Hindi Text (“एक”).

TOOLS AND TECHNOLOGIES USED

To build the model, various tools and technologies were used:

Technologies :

- Deep Learning - Tensorflow, Tflearn
  We used Tensorflow to train the dataset of words using a LSTM (Long Short Term) neural network.

- Audio Preprocessing - NLTK
  NLTK was used to extract frequency and width of audio clips after sampling

- Plotting Graphs - Matplotlib
  Matplotlib was used to plot the Mel Frequency Coefficients in order to perform phoneme classification.

- Programming Language - Python 3.6
  We chose Python because of its extensive support for machine learning and deep learning libraries and relatively easy implementation.

Additional Tools :

PyCharm IDE, Github, Sublime Text
We used PyCharm as the IDE for coding, Sublime Text to configure the properties of the neural network and Github as the repository.

METHODOLOGY

Speech Recognition :

- In this approach we build an LSTM recurrent neural network using the TFLearn high level Tensorflow-based library to train on a labeled dataset of spoken digits. Then we test it on spoken digits.

- We label the Dataset with the help of the extracted features (Frequency and width) from the .wav file and perform classification for phoneme.

- This module is almost complete as there are few errors while training the model due to lack of high-end processor e.g. NVIDIA GTX1080.

Speech Translation:

- We use Neural Machine Translation.
- RNN Encoder–Decoder : It consists of two recurrent neural networks (RNN) that act as an encoder and a decoder pair.
- The encoder maps a variable-length source sequence to a fixed-length vector, and the decoder maps the vector representation back to a variable-length target sequence.

Speech Synthesis :

- We use a TTS engine.
- In this approach we convert raw text containing symbols like numbers and abbreviations into the equivalent of written-out words.
- we assigns phonetic transcriptions to each word .
- Converting the symbolic linguistic representation into sound .

RESULTS

- At the end of the training process, we had a loss (the difference between the actual output and predicted one during testing) of 0.22 and an accuracy score of 0.94.
- Our model was successfully able to recognise individual digits spoken as words (“zero” to “nine”) and provide the corresponding output in Hindi. An example is given below:

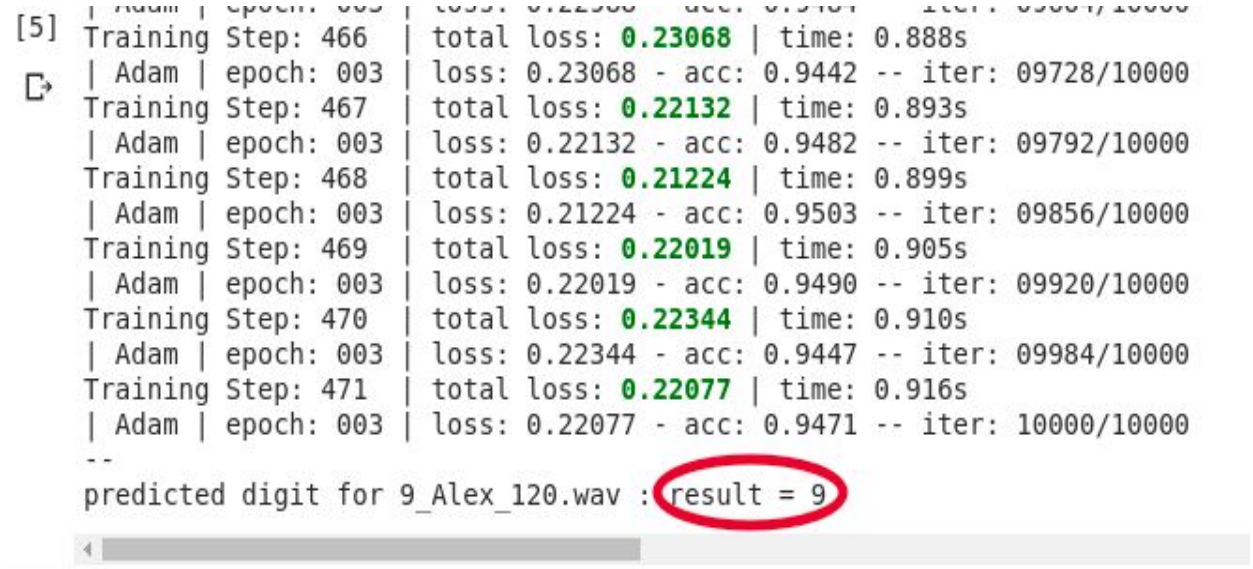

Fig 10.1

OBSERVATION:

STEPS	LOSS	ACCURACY	TIME	TOTAL ACCURACY
119	0.029814	0.998	129	0.997
120	0.033931	0.998	130	0.995
121	0.030113	0.999	131	0.998
122	0.033551	0.998	132	0.997
123	0.027929	0.998	133	0.997
124	0.028464	0.996	135	0.994
125	0.029755	0.997	136	0.997
126	0.029441	0.999	137	0.997
127	0.030021	0.997	138	0.996
128	0.028208	0.998	139	0.998
129	0.029908	0.997	140	0.999
130	0.027707	0.999	141	0.997
131	0.025568	0.999	142	0.996
132	0.029003	0.998	143	0.999
133	0.02222	0.999	144	0.999
134	0.022946	0.999	145	0.998
135	0.029036	0.997	146	0.998
136	0.023717	0.999	147	0.996
137	0.028216	0.996	148	1

Tab 10.1

FUTURE WORK

Implementing the remaining phase - Speech Synthesis:

- Using a TTS(text-to-speech) engine.
- We convert raw text containing symbols like numbers and abbreviations into the equivalent of written-out words.
- We assign phonetic transcriptions to each word .
- Finally, we convert the symbolic linguistic representation into sound .

1. Once we optimize the model for multilingual translation, we plan on using a language with simpler grammatical rules as an intermediate language for optimizing the complexity from O(n^2) to O(2*n).

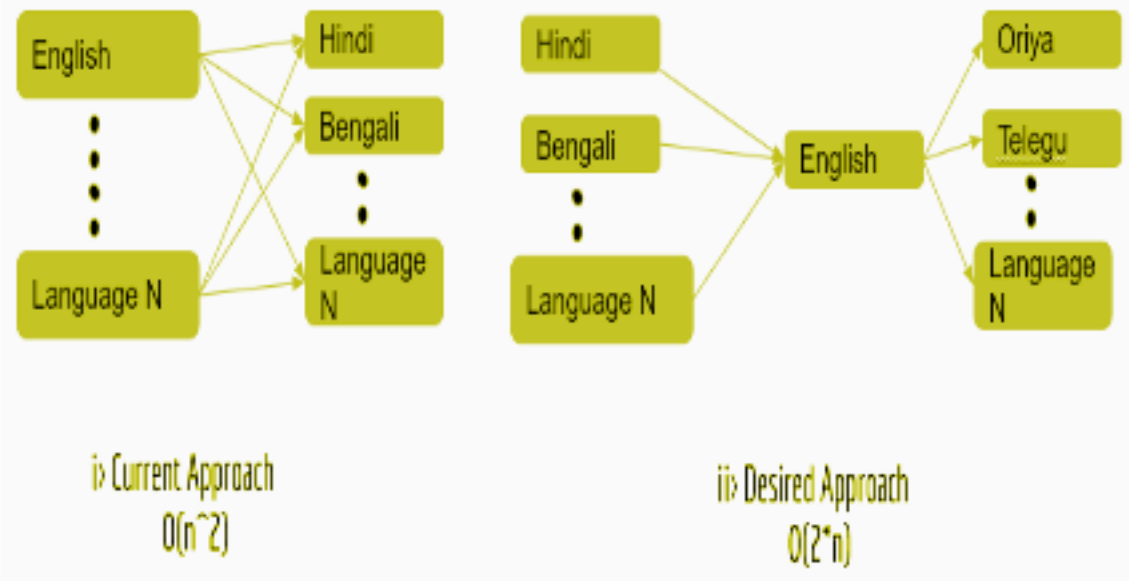

Fig 11.1

2. Using a higher end GPU (eg. NVIDIA GTX 1080) during the training process to facilitate the use of more intensive neural networks with more layers.