End-to-End Speech Recognition Using Recurrent Neural Network (RNN)

Rene Avalloni de Morais* and Baidya Nath Saha
Department of Mathematical and Physical Sciences, Concordia University of Edmonton, Alberta, T5B 4E4

* Corresponding author

doi:https://doi.org/10.21467/proceedings.115.20

ABSTRACT

Deep learning algorithms have received dramatic progress in the area of natural language processing and automatic human speech recognition. However, the accuracy of the deep learning algorithms depends on the amount and quality of the data and training deep models requires high-performance computing resources. In this backdrop, this paper adresses an end-to-end speech recognition system where we fine-tune Mozilla DeepSpeech architecture using two different datasets: LibriSpeech clean dataset and Harvard speech dataset. We train Long Short Term Memory (LSTM) based deep Recurrent Neural Netowrk (RNN) models in Google Colab platform and use their GPU resources. Extensive experimental results demonstrate that Mozilla DeepSpeech model could be fine-tuned for different audio datasets to recognize speeches successfully.

Keywords: Speech Recognition, Mozilla DeepSpeech Model, Deep Learning, Recurrent Neural Networks, Long Short Term Memory (LSTM), Fine-tuning.

I. Introduction

In recent years, Machine Learning (ML), especially Deep Learning (DL) has demonstrated breakthrough successes in various fields, such as speech recognition, natural language processing, and computer vision. Automated speech Recognition systems use trained machine learning algorithms which take human speeches as an input, then interpret, and finally transcribe them into text. The human speech is converted from physical sound to an electrical signal with a microphone, and then to digital data with an analog-to-digital converter. Once digitilized, several models can be used to transcribe the audio data into text format [1]. One of the models that have been widely used by most applications of speech recognition is the static model Hidden Markov Model (HMM), created by Andrey Markov [2]. Automatic Speech Recognition (ASR) can be defined as a technology which facilitates computers to convert the words that a person utter into a readable text through microphone or telephone [3]. An ASR system encounters several challenges associated with different styles, and accent of speaking, environmental noises, and bad quality of audios. To mitigate these issues and improve accuracies, dictionary, language model, and acoustic model are introduced in the process of recognition which are successfully implemented by end-to-end deep learning based approaches [4]. Unlike HMM, deep learning based models take audio features directly as an input and obtains state-of-the-art performances. DeepSpeech model exploits a well-optimized RNN architecture for speech recognition system [5]. Inspired by DeepSpeech model, thhe Mozilla Foundation developed an open-source speech recognition engine called Mozilla DeepSpeech (MDS) which has been fine-tuned for this study. The MDS architecture is an ASR end-to-end extremely parametrizable, which enables to fine-tune the trained speech recognition models using hidden layers through recurrent neural network (RNN) [6] as shown in Figure 1.

Proceedings DOI: 10.21467/proceedings.115; Series: AIJR Proceedings; ISSN: 2582-3922; ISBN: 978-81-947843-5-7

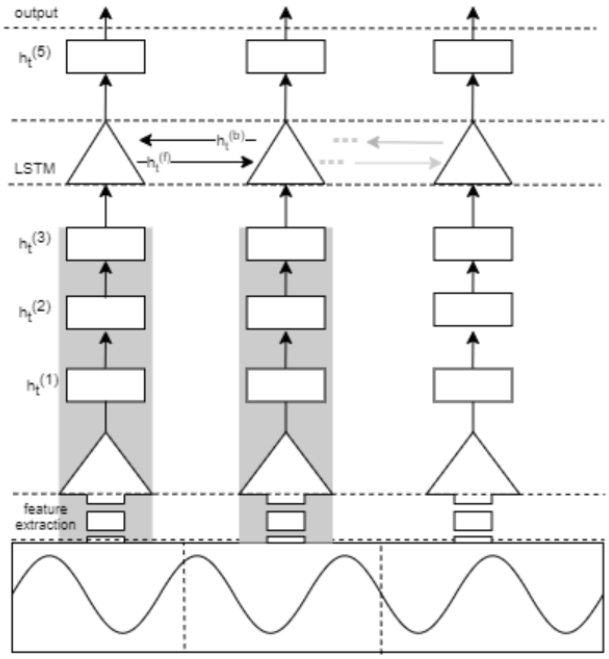

Fig. 1: Structure of RNN model and notation

II. Methodology for Fine-tuning DeepSpeech Model

The proposed work addresses fine-tuning the DeepSpeech engine, which facilitates end-to-end speech recognition for different audio dataset.

We conducted the fine-tuning experiment on two benchmark audio datasets: Harvard dataset and LibriSpeech audio datasets. First, we divide and use 70% as training, 20% as validation, and 10% as test for Harvard dataset and carried out fine-tuning experiment on four different ways:

1. Trained DeepSpeech Model has been used on Harvard dataset for speech recognition evaluation.
2. DeepSpeech Model is fine-tuned by LibriSpeech dataset and then tested on Harvard test dataset.
3. DeepSpeech Model is first fine-tuned by LibriSpeech dataset and then fine-tuned by Harvard training dataset before testing on Harvard test set.
4. DeepSpeech Model is fine-tuned by Harvard training dataset and then tested on Harvard test set.

Harvard validation dataset has been used in the training stage for parameter optimization. Performance of four different fine-tuning strategy of DeepSpeech model for Harvard dataset is illustrated in the experimental results and discussions section.

Figure 2 illustrates the methodology followed in this study to fine-tune the proposed DeepSpeech model.

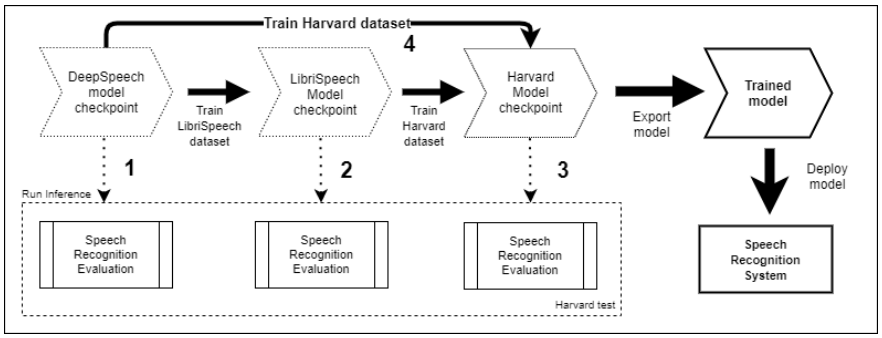

Fig. 2: Fine-tuning process diagram for DeepSpeech Model

III. Experimental Results and Discussions

We conduct the experiment in high-end Google Colab GPU (Tesla T4 GPU card based) which facilitate deep learning model train and inference acceleration. The code is written in Python notebook which facilitates interactive code and data visualization. Data description, speech data processing, computational resources used in this experiment, fine-tuning algorithms, and their performances are discussed below.

A. DATASET

Two datasets, LibriSpeech and DeepSpeech were used in this fine-tuning experiment. The LibriSpeech clean dataset, which is a part of the LibriVox Project dataset, was used for the first training. It contains a training set of 100 hours of 16kHz data of reading English speech from audiobooks [7].

On the other hand, sentences in the Harvard dataset are a collection of phrases widely used in various research fields, such as telecommunications and speech. This dataset was manually split using Audacity (open-source software available for editing sounds) to be used for the testing of the ASR. A total of 10 audio files was downloaded, originally containing 10 sentences each, and split into 100 sentences. out of ten, five files contain woman voice and the remaining five files include the man voice. These files were used to fine-tune the DeepSpeech model to improve accuracy [8].

B. SPEECH DATA PROCESSING

The Harvard dataset and the corresponding text transcription files(one text file contians the text for each audio file) were kept in the same folder. A Linux Bash Script (make_files.sh) was built to prepare and generate the files automatically and all the steps as mentioned below were followed:

(i) Convert all audio data from 8khz (original) to 16khz (frequency supported by DeepSpeech) using the SoX utility;

(ii) Generate a JSON output file containing all the following data attributes: (full path of the split audio file, original text transcription, encoding of audio data, language spoken, and gender of the speaker);

(iii) Generate the three CSV training files for Speech Recognition model training: train.csv, dev.csv, and test.csv;

To develop the ASR (SpeechR.py), code is written in Python to fine-tune the DeepSpeech engine. The code first reads the content of a JSON file and uses its attributes as input to fine-tune the Mozilla DeepSpeech recognition model. This file contains the attributes for all Harvard Sentences (100 audio files and its text transcriptions). Then the function ”speech()” is called to load the model and scorer file used by Mozilla Deepspeech. The frequency of the audio file is validated and afterward, the inference is started. Finally, the function returns the predicted audio-to-text transcription, prints the results to the buffer and a CSV file with the execution timestamp and the content of the audio transcription. TensorFlow framework were used for fine-tuning the DeepSpeech recognition model.

Proceedings DOI: 10.21467/proceedings.115
ISBN: 978-81-947843-5-7

Series: AIJR Proceedings
ISSN: 2582-3922

C. PLATFORM AND SYSTEM ARCHITECTURE

OS libraries	Python libraries
libasound2	deepspeech-gpu==0.9.1
libasound2-dev	tensorflow-gpu==1.15.4
libsox-dev	pysox
sox	setuptools==49.6.0
libsox-dev	pip==20.2.2
libpthread-stubs0-dev	wheel==0.34.2
libgomp1 libstdc++	

TABLE I: Required libraries for training DeepSpeech engine.

The script written for fine-tuning the DeepSpeech engine in the Tensorflow framework uses the libraries mentioned in the Table I for training the model.

Datasets were stored in the Google Drive and the data used for this experiment took 948.8 GB in the memory. The architecture of Virtual Machines for Google Colab environment used in this experiment are as follows: Ubuntu Linux 18.04.5 LTS (Bionic Beaver), Intel(R) Xeon(R) CPU @ 2.20GHz, 2 CPU Cores, 2 GPU Tesla T4, NVIDIA-SMI 455.45.01, Driver Version: 418.67, CUDA Version: 10.1.

D. FINE-TUNING THE DEEPSPEECH MODEL

This experiment involves fine-tuning parameters of the DeepSpeech model since both LibriSpeech and Harvard dataset contain Englsih alphabet on which the Mozilla DeepSpeech model has been trained. The values of a list of parameters that affect the training accuracy such as, the number of hidden neurons, learning rate, number of epochs, dropout rate etc. need to carefully determined. A large number of epochs and dropout rate could cause poor performance for both training and validation. The learning rate of 0.01 or greater could help to converge the algorithm faster at suboptimal solution. However, smaller values of 0.000001 or less, normally makes the cost to change slightly until the iteration ends [9]. Inspired by the previous researches available in the literature, the values of the parameters selected for this study are mentioned in Table II.

Parameter	Value
n_hidden	2048
epoch	10
learning_rate	0.0001
dropout_rate	0.4
train_batch_size	8
dev_batch_size	8
test_batch_size	1
train_cudnn	true

TABLE II: Values of training parameters used in this experiment

It took 48 hours to train LibriSpeech model.

E. EVALUATION METRICS FOR EVALUATING TEXT TRANSCRIPTION

Quantitative evaluation was conducted by using two metrics: Word Error Rate (WER) and Character Error Rate (CER). These metrics calculate the number of words and characters wrongly predicted by the models [7]. WER is calculated by, Word Error Rate (WER) $= \frac{(I+D+S)}{N}$ where, I = Number of word inserted in a sentence, D = Number of words deleted in a sentence, S = Number of words substituted in a sentence, and N = Total number of words in a sentence. CER is computed using the same equation, only instead of words, characters are considered.

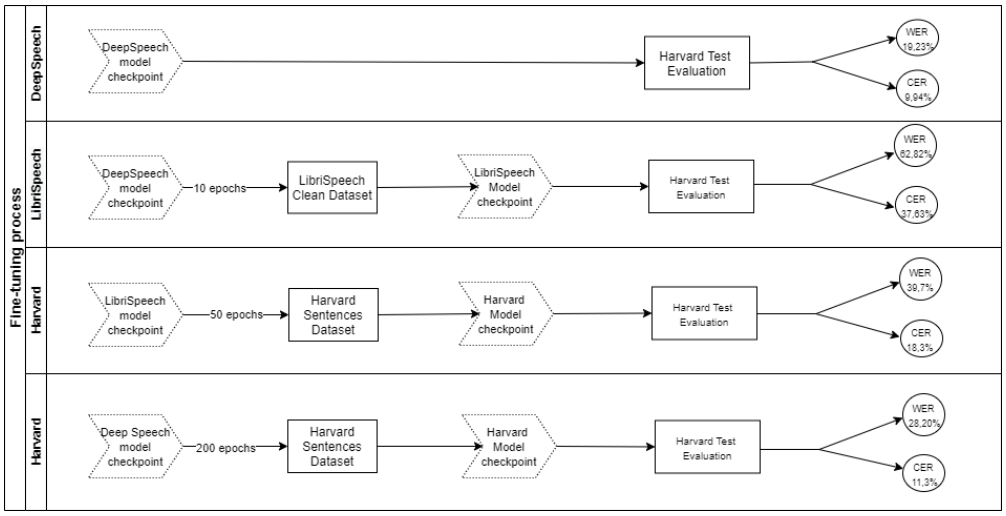

Fig. 3: Four different fine-tuning process for DeepSpeech Model on Harvard dataset

F. Model evaluation

Table III shows the results of the different fine-tuning algorithms as illustrated in Figure 3 in terms of WER and CER. Results demonstrated that DeepSpeech model can successfully transcribe the audio data from Harvard dataset without any fine-tuining. Table IV demonstrates the performance of the DeepSpeech Model on Harvard test set after fine-tuning through Hardvard training set for different epochs. Results show that the model with 200 epochs produces less WER and CER.

Fine tuning Process	WER	CER
DeepSpeech Model (Without any fine tuning)	19.23%	9.94%
Librispeech Model (Fine tuned by Librispeech dataset)	62.82%	37.63%
Harvard Model (Fine tuned by Librispeech dataset first and then with Harvard dataset)	39.70%	18.30%
Harvard Model (Fine tuned by Harvard dataset)	28.20%	11.30%

TABLE III: Results of four different fine-tuning methods for Harvard dataset)

Proceedings DOI: 10.21467/proceedings.115
ISBN: 978-81-947843-5-7

Series: AIJR Proceedings
ISSN: 2582-3922

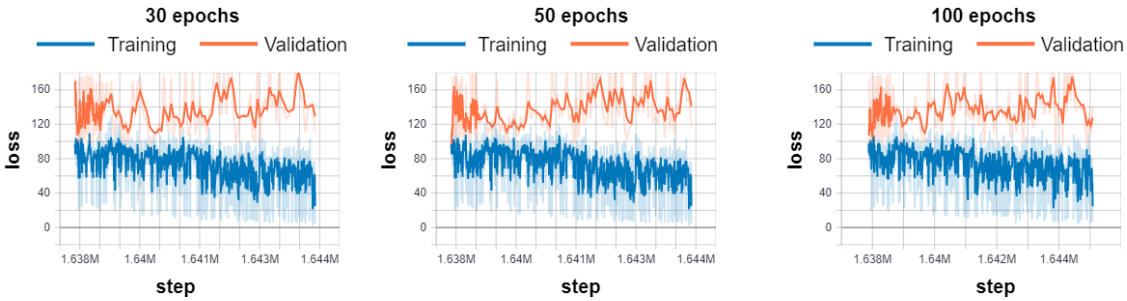

Fig. 4: Training (blue) and validation (orange) loss over steps for Harvard Dataset

Figure 4 illustrates train (blue) and validation (orange) loss over steps for 30, 50, and 100 epochs in the Harvard model fine-tunning. As the number of epochs increases, more times the weight is changed in the neural network. However, the results have not shown improvement, differently than when we trained the model directly using 200 epochs.

Number of epochs	WER	CER
30	44.87%	22.04%
50	44.87%	22.31%
100	43.59%	22.60%
200	42.30%	22.04%
200	**28.20%**	**11.30%**

TABLE IV: Results of DeepSpeech Model on Harvard test set after fine-tuning through Harvard training set

IV. Conclusion and Future Work

This study demonstrates how to fine-tune RNN based Mozilla DeepSpeech model for end-to-end speech recognition. Deep Neural Networks are being widely used for speech recognition, however, we noticed a need to scale speech recognition training for large datasets as it demands high-performance computational resources. We demonstrated the challenges to fine-tune the model and train for a very large dataset (LibriSpeech). We conducted this study in the cloud environment and used all free computational and memory resources provided in Google Colab. Google colab allows training deep learning models with considerably large datasets. This work developed a successful ASR system using Google Colab Notebook. In the future, we would like to implement feature selection strategy for audio features to improve the performance of ASR system.

REFERENCES

[1] N. Goyal and S. S. Rathore, "Predictive Visual Analysis of Speech Data using Machine Learning Algorithms," *Proceedings of 3rd International Conference on Emerging Technologies in Computer Engineering: Machine Learning and Internet of Things, ICETCE 2020*, no. February, pp. 69–73, 2020.

[2] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[3] H. Prakoso, R. Ferdiana, and R. Hartanto, "Indonesian Automatic Speech Recognition system using CMUSphinx toolkit and limited dataset," *2016 International Symposium on Electronics and Smart Devices, ISESD 2016*, pp. 283–286, 2017.

[4] H. H. Nasereddin and A. A. R. Omari, "Classification techniques for automatic speech recognition (ASR) algorithms used with real time speech translation," *Proceedings of Computing Conference 2017*, vol. 2018-Janua, no. July, pp. 200–207, 2018.

[5] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, and A. Y. Ng, "Deep Speech: Scaling up end-to-end speech recognition," pp. 1–12, 2014. [Online]. Available: http://arxiv.org/abs/1412.5567

[6] D. Karkada and V. A. Saletore, "Training Speech Recognition Models on HPC Infrastructure," *Proceedings of MLHPC 2018: Machine Learning in HPC Environments, Held in conjunction with SC 2018: The International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 124–132, 2019.

[7] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2015-August, pp. 5206–5210, 2015.

[8] N. Power, E. Committee, I. Power, and E. Society, "IEEE Recommended Practice for the," vol. 2004, no. June, 2005.

[9] S. Suyanto, A. Arifianto, A. Sirwan, and A. P. Rizaendra, "End-to-End Speech Recognition Models for a Low-Resourced Indonesian Language," *2020 8th International Conference on Information and Communication Technology, ICoICT 2020*, no. ii, 2020.

Proceedings DOI: 10.21467/proceedings.115
ISBN: 978-81-947843-5-7

Series: AIJR Proceedings
ISSN: 2582-3922