Journal of Theoretical and Applied Information Technology
15th February 2024. Vol.102. No 3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AN END-TO-END DEEP LEARNING APPROACH FOR AN INDIAN ENGLISH REPOSITORY

PRATHIBHA SUDHAKARAN[1], ASHWANI KUMAR YADAV[2] , SUNIL KARAMCHANDANI [3]

[1]Assisstant Professor, Muthoot Institute Of Technology and Science,Kochi,Kerala , India
[2]Assistant Professor , ASET,Amity University, Jaipur, Rajasthan, India
[3] Associate Professor, DJ Sanghvi College of Engineering, Mumbai, Maharashtra, India

E-mails: [1]prathibhasudhakaran@mgits.ac.in, [2]ashwaniy2@gmail.com,
[3]skaramchandani@rediffmail.com

ABSTRACT

A voice recognition system that immediately translates raw audio waveforms into text without the need for separate components for language and acoustic modelling or manually constructed feature engineering is known as end-to-end deep learning for continuous speech recognition. It learns the whole audio to text mapping using a single deep neural network model. Conventional speech systems rely on intricate processing pipelines, but this method is far simpler. An end-to-end model in voice recognition is a simple single model that operates directly on words, subwords, or characters and may be trained from the ground up. This simplifies decoding by doing away with the requirement for both explicit phone modelling and a pronunciation lexicon. Deepspeech was used to construct and test the model, which was designed for Indian English. Additionally, a comparison is made between the results of the bi-directional RNN-based system and the traditional HMM model. With our method, we can quickly get a large amount of heterogeneous data for training due to a number of unique data synthesis techniques and an extremely effective RNN development system that utilises several GPUs. The connectionist temporal classification (CTC) objective function is used to infer the alignments between speech and label sequences, obviating the requirement for pre-generated frame labels. Experiments demonstrate that the RNN-based model has been observed to have equal word error rates (WERs) while also significantly speeding up the decoding process when compared to traditional hybrid HMM based on Kaldi.
Keywords: *End-To-End,HMM,CTC,RNN,CSR, Indian English, Deep Speech*

1 INTRODUCTION

The tasks involved in the field of speech recognition is difficult and complex. Teaching machines to comprehend and interpret spoken language is a huge technological achievement, despite the ease with which humans can do so. Speech variability, background noise, vocabulary and language models, contextual understanding, dependent and independent speaker models, and data availability and quality are some of the aspects that make speech recognition difficult. Despite these obstacles, recent developments in machine learning, deep learning, and neural network topologies have allowed for a considerable breakthrough in voice recognition technology. While attention approaches allow the model to focus on salient regions of the input signal, transformers facilitate the description of long-range relationships within the signal. These developments have greatly improved speech processing systems' functionality and performance, creating new prospects for use in a variety of industries.[1]

Despite significant advancements in speech processing, deep learning still has some issues that need to be resolved. These difficulties include the need for large amounts of labeled data, the models' interpretability, and their resilience to various
environmental factors. Deep learning approaches have significantly improved the analysis, synthesis, and detection of speech signals, which are all parts of speech processing. Many researchers previous workhave shed light on the

potential of deep learning for addressing the current issues and furthering speech processing research by analyzing the current state-of-the-art methodologies.

In recent years Deep learning techniques, have aroused a lot of attention as a method to create hierarchical representations from unlabeled data. Recent developments in algorithms and computer technology have enabled end-to- end training of neural networks for tasks that previously needed significant human knowledge. Deep learning algorithms for audio data have not been thoroughly investigated. We deploy recurrent deep belief networks to audio data in this paper and test them on pretrained Mozilla model. We show that learnt features match to phones/phonemes in the case of speech data. Recent papers have shown that the RNN based training systems do not need a dictionary. It has been proved as a benchmark performance in a noisy environment. [2] A system based on a combination of deep bidirectional LSTM recurrent neural network architecture and CTC function have proved to reduce the word error rates considerably. [3] Furthermore, we calculate the word error rate and character error rate with and without the language and dictionary. We hope that this paper will spark additional research into deep learning algorithms for a variety of audio recognition tasks.

2. RECURRENT NEURAL NETWORK

Our approach is based on a recurrent neural network that has been trained to process voice spectrograms and produce text transcriptions. Assign a label y to a training set that will be sampled for a single utterance (x). Every utterance is a time series in which every time slice has a vector of audio properties. [4] The purpose of RNN is to turn a sequence of input characters into a sequence of character probabilities for transcription with $\hat{y}(t) = p(C_t|x)$.……(2.1)

Our RNN is made up of five levels of concealed units. The first three layers do not repeat themselves. For the first layer, the output is determined by the spectrogram frame xt and a context of C frames on either side at each time t. For each time step, the remaining non-recurrent layers act on independent data. As a result, the first three layers are computedfor each time 't' by:

$$h^{t}(t) = g\left(w^{l}h_{t}^{(l-1)} + b^{(l)}\right)$$ ……(2.2) where g(z) is the clipped rectified -linear activation function and w and b are the weight matrix and the bias parameters for layer 1.

[4] The fourth layer includes 2 sets of hidden units: a set with forward recurrence and a set with backwardrecurrence

$$h_t^{(f)} = g\left(w^{(4)}h_t^{(3)} + w_h^{(f)}h_{t-1}^{(f)} + b^{(4)}\right)$$
……..(2.3)

$$h_t^{(b)} = g\left(W^{(4)}h_t^{(3)} + W_r^{(b)}h_{t+1}^{(b)} + b^{(4)}\right)$$
……..(2.4)

The inputs to the fifth layer are both forward and backward units. The output layer is a conventional softmax functionthat returns the anticipated character probabilities for each time slice t and alphabet character k.

$$h_{t,k}^{(6)} = \hat{y}_{t,k} = P(c_t = k|x) = \frac{\exp\left(W_k^{(6)}h_t^{(5)} + b_k^{(6)}\right)}{\Sigma_j \exp\left(W_j^{(6)}h_t^{(5)} + b_j^{(6)}\right)}$$
...(2.4)

where Wk and bk denote the kth column of the weight matrix

and kth bias. [2]To measure the error in prediction, we compute CTC loss after we compute prediction.[5] We can evaluate the gradient with respect to the gradient outputs given the ground-truth character sequence y during training, and then compute the gradient with respect to all of the model parameters using back propagation across the rest of the network. For training, we employ Nesterov's accelerated gradient approach.[6]

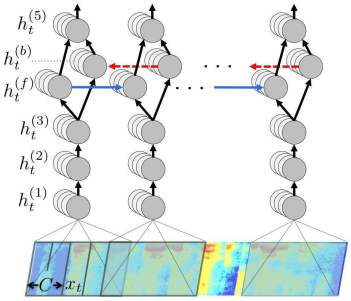
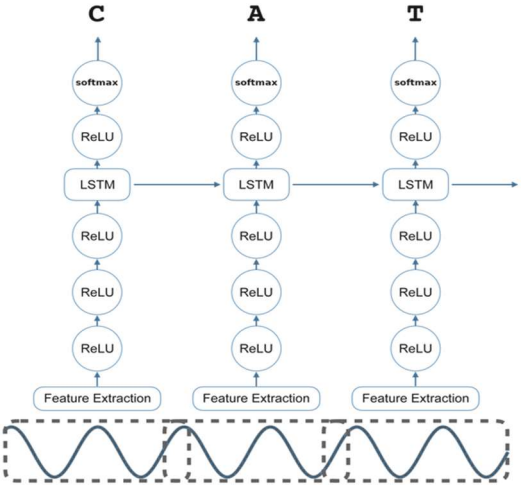


Figure 1 : Structure of the RNN model[2]

Unlike kaldi-based models, we just utilize a single recurrent layer and do not require LSTM. The LSTM system has the drawback of requiring numerous gating neuron responses to be computed

and stored at each stage. This modest additional cost can constitute a computing bottleneck since the forward and backward recurrences are sequential.[3]

Figure 2: The Complete RNN Model



Each of the MFCC characteristics of a time-slice of the speech sample is represented by a vector of at most n steps vectors. The amount of MFCC features will be determined by the data set's sampling rate. If the sample rate is 8kHz, we employ 13 features in general. The MFCC characteristics of a particular time-slice are not simply passed to RNN. A context of C frames on each side of the frame in question is also provided to it. The variable n context stores the number of frames in this context.

3. CONNECTIONIST TEMPORAL CLASSIFICATION

We use the CTC loss function to train neural networks to achieve maximum likelihood training of letter sequences with auditory information as input. CTC is a generic loss function that can be used to train systems on sequence problems where the input and output sequences are not aligned. CTC models possible re-orderings but does not account for time warping of the output sequence relative to the input sequence.

When working with voice recognition, re-ordering is a difficulty in machine translation, but it isn't an issue because our transcripts offer the exact order in which words appear in the input audio. Connectionist Temporal Classification is an objective function that enables an RNN to be trained for sequence transcription tasks without the input and target sequences being aligned beforehand.[3] Each of the transcription labels (characters, phonemes, musical notes, etc.) has its own unit in the output layer, as well as a 'blank' unit that refers to a null emission.

Bridle (1990) defined a CTC network as having a softmax output layer with one more unit than the number of labels in L. The probability of seeing the appropriate labels at certain times are understood as the activations of the initial |L| units. The likelihood of seeing a 'blank,' or no label, activates the additional unit. These outputs describe the probability of all potential label sequence alignments with the input sequence when combined. The overall probability of any label sequence is then calculated by adding the probabilities of its various alignments.

For an input ‘x’ of length ‘T’ we can define a RNN with ‘m’ inputs , ‘n’ outputs and ‘w’ weights as a continuous map Nw:

(Rm)T → (Rn)T……..(3.1)

Let y = Nw(x) be the sequence of network outputs and weight ‘w’ vector , is the activation function of the output unit ‘k’ attime ‘t’, then the probability distribution over the set L’T of length T sequences over the alphabet L’ = LỤ{blank}

$$p(\pi|x) = \prod_{t=1}^{T} y^{t}_{\pi_t}, \forall \Pi \in L'^{T} \quad \text{……..(3.2)}$$

The next step is to define a many to one map , lets say B. We do this simply by removing the blanks and repeated labels from the paths. E.g B(a_ _ g_ hh_) = agh. This is used to define the conditional proability of a given labelling 1εL.

$$p(l \mid x) = \sum_{\pi \in B^{-1}} p(\pi|x) \quad \text{……..(3.3)}$$

while constructing the classifier output of the classifier should be the most probable labelling for the input sequence: h(x) = arg max p (l|x). This task of finding the labelling is known as decoding.

4. LANGUAGE MODEL

The RNN model can learn to produce understandable character-level transcriptions when trained with huge amounts of labelled speech data. Many of the errors that occur in our model are proper nouns. Below is an example from the model. The speakers are reading from a book author of

danger trail, Philip steel. The speakers are both male and female from different backgrounds whose dialects impact the English rendered.

Table 1:Original Transcript And The RNN Output

Original Transcript	RNN output
I was about to do this when cooler judgment prevailed	i was about to do this when cooler judgment proved
why should we bother ourselvesabout this lion	why should we bother ourselvesabout this line

Table 2: Original Transcript And The RNN Output

Original Transcript	RNN output
I am in search of a shelter the huntercannot reach	I am in search of a shelter the hunterin a reach
this third part begins with a verse trust not even a close friend who earlier was your enemy	this dart but begins with the worsttrust not even a close friend who only a was your any me

We search for the sequence of letters c1, c2,... that is most likely based on both the RNN output and the language model using the output P(c|x) of our RNN (where the language model interprets the string of characters as words).[7] In particular, we're looking for a sequence c that optimizes the combined goal.

$$Q(c) = log(P(c|x) + \alpha log(Plm(c)) + \beta word\_cou$$

Where α and β are tunable parameters.
'Plm' is the probability of the sequence 'c' as per the N-gram model. [7] In this paper experiments performed are with and without the language model.

5. OPTIMIZATIONS

Here we are using Adam method of optimization, a first-order gradient-based optimization technique similarly we also observe in a paper by Sutskever et.al [10]

When we use multiple GPU we use model parallelization, so each GPU has a specific model. When one runs the model one GPU does back propagation. With that out of the way, we can introduce a tower for each GPU and compute and report the optimization gradients and average loss across towers for each batch. To return and aggregate any non-finite files in the batches, the mean of the losses is determined along with tower gradients.

Data parallelism improves training time for small multiples of the minibatch size (e.g., 2 to 4), but the training convergence rate decreases as more samples are batch into a single gradient update. That is, processing twice as many instances on twice as many GPUs doesn't result in a training speedup of 2. Fixing the overall minibatch size but spreading the samples across two GPUs is likewise

for stochastic objective functions based on adaptive lower-order moment estimations. The approach is simple to develop, computationally efficient, requires minimal memory, is invariant to gradient diagonal rescaling, and is ideally suited for issues with huge amounts of data and/or parameters.
Non-stationary targets and situations with highly noisy and/or sparse gradients may also benefit from the strategy. The hyper-parameters have straightforward meanings and need minimal adjustment in most cases. [8].Individual adaptive learning rates for various parameters are calculated using estimations of the first and second moments of the gradients; the name Adam comes from adaptive moment estimation.

5.1 Data and Model Parallelism

To successfully utilize multiple GPUs, additional abstractions must be introduced that are not present while using a single GPU and that assist the multiple- GPU use case. In particular, a way to segregate the inference and gradient computations on the individual GPUs must be introduced. A 'tower' is the abstraction we propose for this purpose. Two qualities define a tower: the first is Scope. The function 'tf.name scope()' is used to isolate actions inside a tower. All
activities inside 'tower 0', for example, may be prefixed with 'tower 0/'. The second is Device, which is a hardware device given by 'tf.device()' and on which all tower operations run. All operations of 'tower 0', for example, might run on the first GPU 'tf.device('/gpu:0').
When utterances have variable durations, data parallelism is difficult to accomplish because they cannot be integrated into a single matrix multiplication. We tackle the issue by selecting our training samples by length and merging only utterances of comparable length into minibatches, padding with silence when needed to ensure that all utterances in a batch have the same length. The ITPACK/ELLPACK sparse matrix format inspired this idea.[9]

inefficient since most operations become memory-bandwidth constrained when the minibatch size inside each GPU diminishes. By splitting the model, we can scale the model even more.[11][12]

5.2 Striding

We attempted to reduce the running time of our RNN's recurrent layers, which are the most difficult to parallelize. Finally, we reduce the length of the recurrent layers by removing "steps" (or strides) of size 2 from the original input, resulting in an unrolled RNN with half as many steps. This is comparable to a convolutional network in the first layer with a step-size of 2.[13] Our dataset has been provided by IITM for Indian English. The wall street journal, switchboard and the fischer corpora are all published by the linguistic data consortium. [14]

Table3 : Summary Of All The Datasets Used To Train Deep Speech [14]

Data Set	Type	Hours	Speakers
WSJ	Read	80	280
Switchboard	Conversational	300	4000
Fisher	Conversational	2000	23000
Baidu	Read	5000	9600

6. RELATED WORK

Several aspects of our study are based on earlier findings. In the early 1990s, neural network acoustic models and other connectionist methods were initially applied to speech pipelines.[15][16][17]. Graves et al. [18]earlier established the "Connectionist Temporal Classification" (CTC) loss function for grading RNN transcriptions and previously used this technique to speech using LSTM networks. [3]We employ the CTC loss for part of our training, but we use considerably simpler recurrent networks with rectified-linear activations instead. [19][20][21]. This experiment is comparable to Hannum et al [7] bidirectional RNN, but with some adjustments to improve its scalability. We demonstrated that these simpler networks may be successful even without the more complicated LSTM machinery by concentrating on scalability.
In [3] the system is based on a combination of the deep bi-directional LSTM recurrent neural network architecture and the CTC objective function. The system achieved an error rate of 27.3% on the WSJ corpus with no prior linguistic information, 21.9% with only a lexicon of allowed words and 8.2% with a trigram language model. They concluded that character level speech transcription can be performed by RNN with minimal pre-processing and no explicit phonetic representation. In [2] the key approach is a well optimized RNN training system without the need for a phonetic dictionary that uses multiple GPU's as well as set of novel data synthesis techniques. It has been a benchmark in noisy environments' [22]. G.Dahl et.al have described the method to apply CD-DNN- HMMs to LVSR (large vocabulary speech recognition) that analyses the effects of various modeling choices on performance. An absolute SER of 5.8% and 9.2% over the CD-GMM-HMMs trained using MPE and ML criteria. In [23] J.Dean et.al have developed a software framework called DistBelief that can utilize computing clusters with thousands of machines to train large models. D.Ellis et.al [17] the network architecture comprised of a single large hidden layer where doubling the training data and system has provided diminishing returns of error rate reduction. A.Graves et.al [18] find that CTC is suitable for tasks like keyword spotting where segme ntation is required. CTC implicitly models inter-label dependencies as well. In [4] Mike Schuster et.al researched that BRNN may be trained without being restricted to just utilising input data up to a predetermined future frame. This is achieved by concurrently training it in both positive and negative time directions. I.Sutskever et.al in [10] have shown that LSTMs trained on reversed source sentences performed much better on lengthy sentences than LSTMs trained on raw source sentences, suggesting that reversing the input phrases improves memory use. They used LSTMs with 4 layers and stated that deep LSTMs outperformed shallow LSTMs.
In [24] Y.Miao et.al used Eesen framework and concluded that Acoustic modelling in Eesen entails training a single recurrent neural network (RNN) to predict context-independent targets (phonemes or characters). We use the connectionist temporal classification (CTC) objective function to infer the alignments between speech and label sequences, which eliminates the necessity for pre-generated frame labels. L.Lu et.al obtained promising identification accuracy without using an explicit language model or pronunciation lexicon, showing that this technique deserves further exploration.

The use of coupled BiRNNs in the encoder is critical to the success of this model architecture, and employing distinct feedforward neural networks for feature extraction in the encoder may lower the word error rate. Reducing the durations of input sequences makes RNN training simpler, which increases recognition accuracy while also speeding up the training process.[25] T.Mikolov et.al in [26] were able to create a topic-conditioned RNNLM by doing Latent Dirichlet Allocation utilising a block of previous text. The main benefit of this technique is that it avoids the data fragmentation that comes with developing several topic models on distinct data subsets. The primary concept is to use a continuous space representation of the preceding words and sentences to condition the hidden and output vectors. One may avoid the data fragmentation associated with the usual method of developing several topic-specific language models by using a representation based on Latent Dirichlet Allocation.

The importance of scalability in deep learning has been extensively researched, [27][[28]and the utilization of parallel processors (particularly GPUs) has been critical to recent large-scale DL achievements.[28][10]. To obtain even greater efficiency, researchers have started to choose architectures that transfer well to GPU hardware, such as convolutional [29][30][31]and locally linked networks[32] particularly when optimized libraries like cuDNN [33]and BLAS are available. Indeed, leveraging high performance, computing infrastructure, neural networks with more than 10 billion connections can now be trained using GPU clusters.[11] These outcomes from the various research related work prompted us to prioritize scalable design decisions that would allow us to efficiently use many GPUs before attempting to create the algorithms and models themselves. As huge labelled training sets have been utilized to feed bigger and larger DL systems, considerable gains in performance have been made.[34][30]. Huge standard datasets, like the Fisher corpus[14]which has 2000 hours of recorded speech, are uncommon and have just lately been explored. We depend on both vast collections of labelled utterances and synthesis approaches to produce unique instances to fully use the interpretive capacity of the recurrent networks at our disposal. This method is well-known in computer vision, but we've found it to be very useful and successful for speech when correctly implemented.

Using a well-designed random initialization and a certain sort of slowly growing schedule for the momentum parameter, stochastic gradient descent with momentum may train both DNNs and RNNs (on datasets with long- term dependencies) to levels of performance previously only possible with Hessian-Free optimization.[6] In [35] The authors provide char2Wav, a system that consists of two parts: a reader and a neural vocoder. With attention, the reader is an encoder-decoder model. The encoder is a bidirectional recurrent neural network (RNN) that receives text or phonemes as inputs and creates vocoder acoustic characteristics, while the decoder is a recurrent neural network (RNN) with attention. A conditional expansion of Sample RNN, neural vocoder creates raw waveform samples from intermediate representations. Unlike typical voice synthesis algorithms, Char2Wav learns to generate sounds from text.

X.Jin et.al [36] experimented and reveal that in the the training of acoustic models, the models trained by GMM- HMM are used as baselines and inputs, then BLSTM-RNN and TDNN are used for training respectively. After that another 3 acoustic models are generated through transfer learning. ally the test set is decoded and each model is evaluated separately. The experimental results are evaluated and analyzed by WERs. According to the experimental results, when dealing with a low-resourced language such as Indian English, PLP is recommended for feature extraction and BLSTM-RNN is recommended for acoustic model training. In industrial applications W.Xiong et.al in [37] have discussed Microsoft's 2016 conversational voice recognition technology.

The application of CNNs in the acoustic model, as well as RNN language models, has shown to be quite useful. On the NIST 2000 Switchboard set, our best single system has an error rate of 6.9%.On the Switchboard test data, an ensemble of acoustic models improves the state of the art by 6.3 percent. In [38] K.Li et.al have build a cache model and a deep neural network on conversational transcriptions and found 3.9% relative WER reduction and 10.5% PPL reduction on the eval2000 dataset and obtained a 5.6% WER reduction on the subset of eval2000. H.Soltau in [39] have shown that CTC word models perform very well as an end-to-end all-neural speech recognition model without the need of standard context-dependent sub-word phone mes that require a pronunciation lexicon, and without any language model that eliminates the need to decode.A strong, more sophisticated, state-of-the-art baseline with sub-word units performs better

than the CTC word models. On a tough YouTube video transcription job, the final system outperforms a well-trained, traditional context-dependent phone-based system, with a 13.4% word error rate.

In [40] W.Chan et.al have developed a model that learns all components of a speech recognizer simultaneously. The encoder in LAS features a pyramid structure, which lowers the amount of timesteps the decoder needs to deal with. The listener is a pyramidal recurrent network encoder that takes as input the spectra of the filter bank. The speller is a character-emitting recurrent network decoder that is dependent on attention.

7. EXPERIMENTAL SETUP

We have compared our system to the above mentioned related work as reference on an Indian English dataset spoken by both male and female speakers. The dataset consists of 45,000 tokens. WER(word error rate) and CER (character error rate) has been processed in Jupyter notebook in anaconda enviroment. We use a basic kind of speaker adaptation that involves normalising spectral data per speaker.The deepspeech model consists of 5 dense layers and one unidirectional RNN layer, a total of 6 layers in all which is novel to this dataset.

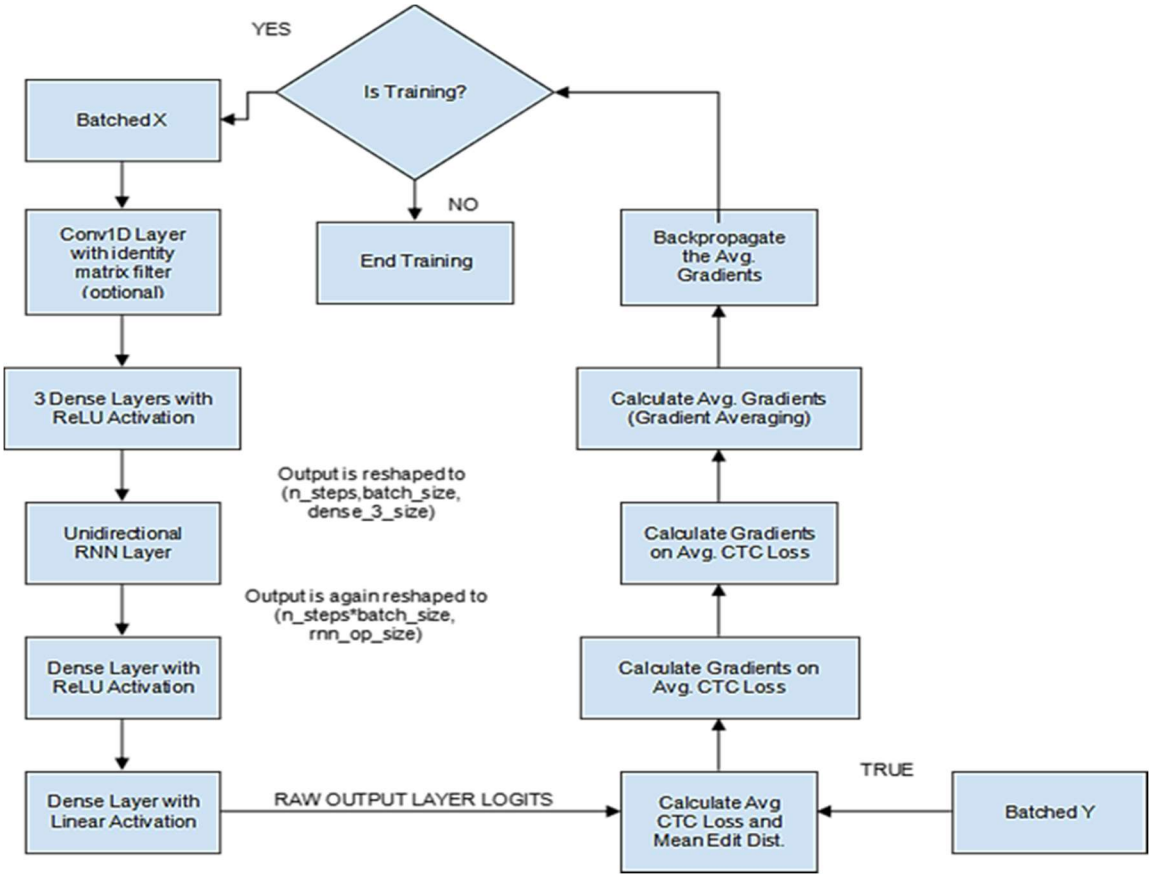


Figure 3: Flowchart Of The Unidirectional RNN Based Deepspeech Dataflow Implemented

7.1 Metrics

WER (word error rate) is defined as the edit/levenshtein distance on word level divided by the amount of words in the original text.

In case of the original having more words (N) than the result and both being totally different, the WER will always be 1 (N/N =1). It has been computed as follows, similarly CER has been computed.

WER = sum(s.word_distance for s in sample)/ sum(s.word_length for s in samples)
CER = sum(s.char_distance for s in sample)/ sum(s.char_length for s in samples)

Word error rate is the most common metric used today to evaluate the effectiveness of an automatic speech recognition system (ASR).

Product developers and data scientists while evaluating speech recognition APIs consider WER as a reliable evaluating system. It is simply calculated as[41]

$$WER = S+D+I /N \quad \text{……..(7.1)}$$

where S = S stands replacing a word,I stands for inserting a word ,D stands for omitting a word ,N is the number of words that were actually spoken.

From our experiments, let us see an example. We observe that 'I' has been substituted by 'the' The following words have been spelled correctly.
Example 1:

mh1045 *ref* i brought this flock with me from there

mh1045 *hyp* the brought this flock with me fromthere.

mh1045 op:

S C C C C C C C
The brought this flock with me from there

$$CER = S+D+I/(S+D+C)\text{……(7.2)}$$

where **S** is the number of substitutions
D is the number of deletions
I is the number of insertions
C is the number of correct characters
N is the number of characters in the reference.[42]

The minimum string distance (MSD) between the provided and transcribed text strings is calculated in character errorr. This is the smallest number of basic operations (substitutions, insertions, and deletions) required to convert one string to another.

The MSD statistic represents the amount of mistakes made by the user when inputting the given text string.

The WER and CER obtained are shown in the table below. We have computed separately for male and female speakers.

Table4 : WER (Word Error Rate) And CER(Character Error Rate) Obtained Through Analysis

Dataset	WER	1-WER	CER	1-CER
Male speakers	0.1721	0.8278	0.0857	0.9142
Female speakers	0.1937	0.8062	0.0982	0.9017

7.2 Discussions

The network must not only learn how to detect speech sounds, but also how to convert them into letters, in order to offer character-level transcriptions. To put it another way, it needs to learn how to spell. This is difficult, particularly in a language with orthographic irregularities like English. The following samples from the evaluation set, decoded without the use of a lexicon or language model, show how the network works:
Original: now you're coming down to business phil he exclaimed.
Output: now you're coming down to business phil he exclaimed.
Original : in the picture he saw each moment a greater resemblance to jeanne.
Output: in the picture he saw each woman a greater resemblance to gene.
Original :its alright thieves first let me know about your expertise in the art and then decide.
Output: is all right dives first let me now about your expertise in the art and then decide.
Original: all of us are trapped in this net because of our weakness for
Output: on furs are trapped in this net because of our weakness for food.
The network, like other voice recognition systems, makes phonetic errors, such as saying 'each lady' instead of 'each moment,' and sometimes mixes homophones like 'hear'-here,' 'hour-are,' and so on. The second issue may be more difficult to solve using a language model than typical, since words that sound similar might spell differently. Unlike phonetic systems, the network produces lexical mistakes, such as mistaking 'knight' for 'night,' and errors that mix the two, such as mistaking 'alright' for 'all right.'
It can accurately transcribe fairly complicated

terms that occur regularly, such as'salutations,' 'possibility,' and'shuffled,' but it has trouble with the sound and spelling of new words, particularly proper names like 'Tanaliraman' and 'harness.' This shows that even in the absence of a dictionary, out-of-vocabulary terms may be a barrier for character-level recognition. The fact that the network can spell at all indicates that it can infer important linguistic information from the training transcripts, opening the door for a true end-to-end voice recognition system.

8. CONCLUSION

This paper shows that a recurrent neural network can perform character-level voice transcription with little pre-processing and no explicit phonetic representation. We have also shown how to combine the network outputs with a language model during decoding using an unique objective function that enables the network to be directly optimised for word error rate. Finally, we obtained state-of-the-art accuracy for speaker independent identification on the Indian English dataset. In the future, it would be fascinating to apply the system to datasets where the language model is less important, such as spontaneous speech, or when the training set is big enough for the network to develop a language model just from the transcripts. Integration of the language model into the CTC or anticipated transcription loss goal functions during training is another potential avenue.

REFERENCES:

[1] Mehrish, A., Majumder, N., Bharadwaj, R., Mihalcea, R. and Poria, S., 2023. *A Review of deep learning techniques for speech processing. Information Fusion*, p.101869.

[2] A. Graves and N. Jaitly, *"Towards end-to-end speech recognition with recurrent neural networks*," 31st Int. Conf. Mach. Learn. ICML 2014, vol. 5, pp. 3771–3779, 2014.

[3] B. Iung, "Cœur et grossesse," EMC - Trait. médecine AKOS, vol. 8, no. 2, pp. 1–4, 2013, doi: 10.1016/s1634- 6939(13)59289-1.

[4] W. J. L. Adams and D. L. Saaty, *"CTC end-to-end,"* 2006.

[5] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, *"On the importance of initialization and momentum in deep learning*," 30th Int. Conf. Mach. Learn. ICML 2013, no. PART 3, pp. 2176–2184, 2013.

[6] A. Y. Hannun, A. L. Maas, D. Jurafsky, and A. Y. Ng, *"First-Pass Large Vocabulary Continuous Speech Recognition using Bi-Directional Recurrent DNNs,"* pp.1–7,2014.Available: http://arxiv.org/abs/1408.2873.

[7] D. P. Kingma and J. L. Ba, *"Adam: A method for stochastic optimization,"* 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc., pp. 1–15, 2015.

[8] Li K, Xu H, Wang Y, Povey D, Khudanpur S. "*Recurrent neural network language model adaptation for conversational speech recognition*" InInterspeech 2018 Sep 2 (Vol. 2018, pp. 3373-3377).

[9] "TR-91-32.pdf." .

[10] I. Sutskever, O. Vinyals, and Q. V. Le, *"Sequence to sequence learning with neural networks,"* Adv. Neural Inf. Process. Syst., vol. 4, no. January, pp. 3104–3112, 2014.

[11] A. Coates, B. Huval, T. Wang, D. J. Wu, A. Y. Ng, and B. Catanzaro,30th Int. Conf. Mach. Learn. ICML 2013, no. PART 3, pp. 2374–2382, 2013.

[12] A. S. MONIN and A. M. YAGLOM, "*Statistical Fluid Mechanics*. Volume 2. Mechanics of Turbulence.," no. (1975), pp. 1–9, 1975.

[13] Y. LeCun et al., "*Backpropagation applied to digit recognition*," Neural computation, vol. 1, no. 4. pp. 541–551,1989,[Online].Available: https://www.ics.uci.edu/~welling/teaching/273ASpring09/lecun-89e.pdf.

[14] C. Cieri, D. Miller, and K. Walker, "*The fisher corpus: A resource for the next generations of speech-to-text*,"Proc. 4th Int. Conf. Lang. Resour. Eval. Lr. 2004, pp. 69–71, 2004.

[15] H.~Bourlard and N.~Morgan, Connectionist Speech Recognition---A Hybrid Approach. 1994.

[16] S. Renals, N. Morgan, H. Bourlard, M. Cohen, and H. Franco, "*Connectionist Probability Estimators In Hmm Speech Recognition*," IEEE Trans. Speech Audio Process., vol. 2, no. 1, pp. 161–174, 1994, doi: 10.1109/89.260359.

[17] D. Ellis and N. Morgan, "*Size matters: An empirical study of neural network training for large vocabulary continuous speech recognition*," ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc., vol. 2, pp. 1013–1016, 1999, doi: 10.1109/icassp.1999.759875.

[18] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber,"*Connectionist temporal classification: Labelling unsegmented*

sequence data with recurrent neural networks," ACM Int. Conf. Proceeding Ser., vol. 148, pp. 369– 376, 2006, doi: 10.1145/1143844.1143891.

[19] H. Li, J. Wang, M. Tang, and X. Li, "*Polarization-dependent effects of an Airy beam due to the spin-orbit coupling*," J. Opt. Soc. Am. A Opt. Image Sci. Vis., vol. 34, no. 7, pp. 1114–1118, 2017, doi: 10.1002/ecs2.1832.

[20] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "*Rectifier nonlinearities improve neural network acoustic models*,"ICML Work. Deep Learn. Audio, Speech Lang. Process., vol. 28, 2013.

[21] M. J. Brown, L. A. Hutchinson, M. J. Rainbow, K. J. Deluzio, and A. R. De Asha, "*A comparison of self- selected walking speeds and walking speed variability when data are collected during repeated discrete trials and during continuous walking*," J. Appl. Biomech., vol. 33, no. 5, pp. 384–387, 2017, doi: 10.1123/jab.2016-0355.

[22] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "*Context-dependent pre-trained deep neural networks for large- vocabulary speech recognition*," IEEE Trans. Audio, Speech Lang. Process., vol. 20, no.1,pp.3042,2012,doi10.1109/TASL.2011.2134090

[23] J. Dean et al., "*Large scale distributed deep networks,*" Adv. Neural Inf. Process. Syst., vol. 2, pp. 1223–1231, 2012.

[24] Y. Miao, M. Gowayyed, and F. Metze, "*EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding*," 2015 IEEE Work. Autom. Speech Recognit. Understanding, ASRU 2015 - Proc., pp. 167–174, 2016, doi: 10.1109/ASRU.2015.7404790.

[25] L. Lu, X. Zhang, K. Cho, and S. Renals, "*A study of the recurrent neural network encoder-decoder for large vocabulary speech recognition,*" Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH, vol. 2015-Janua, pp. 3249–3253, 2015.

[26] T. Mikolov and G. Zweig, "*Context dependent recurrent neural network language model*," 2012 IEEE Work. Spok. Lang. Technol. SLT 2012 - Proc., pp. 234–239, 2012, doi: 10.1109/SLT.2012.6424228.

[27] A. Coates, H. Lee, and A. Y. Ng, "*An analysis of single-layer networks in unsupervised feature learning*," J. Mach. Learn. Res., vol. 15, pp. 215–223, 2011.

[28] Q. V. Le, "*Building high-level features using large scale unsupervised learning*," ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc., pp. 8595–8598,2013,doi10.1109/ICASSP.2013.6639343.

[29] D. C. Cires, U. Meier, J. Masci, and L. M. Gambardella, "*Flexible, High Performance Convolutional Neural Networks for Image Classification*," Proc. Twenty-Second Int. Jt. Conf. Artif. Intell. Flex., pp. 1237–1242, 2013, [Online]. Available: https://www.aaai.org/ocs/index.php/IJCAI/IJCAI11/paper/viewFile/3098/3425.

[30] T. F. Gonzalez, "*Handbook of approximation algorithms and metaheuristics*," Handb. Approx. Algorithms Metaheuristics, pp. 1–1432, 2007, doi: 10.1201/9781420010749.

[31]T. N. Sainath et al., "*Improvements to deep convolutional neural networks for LVCSR*," 2013 IEEE Work. Autom. Speech Recognit. Understanding, ASRU 2013 - Proc., pp. 315–320, 2013, doi: 10.1109/ASRU.2013.6707749.

[32] D. Ciregan, U. Meier, and J. Schmidhuber, "*Multi-column deep neural networks for image classification*," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., no. February, pp. 3642–3649, 2012, doi: 10.1109/CVPR.2012.6248110.

[33] S. Chetlur et al., "*cuDNN: Efficient Primitives for Deep Learning*," pp. 1–9, 2014, [Online]. Available: http://arxiv.org/abs/1410.0759.

[34] G. Zeng, Y. He, Z. Yu, X. Yang, R. Yang, and L. Zhang, "*Preparation of novel high copper ions removal membranes by embedding organosilane-functionalized multi-walled carbon nanotube*," J. Chem. Technol. Biotechnol., vol. 91, no. 8, pp. 2322–2330, 2016, doi: 10.1002/jctb.4820.

[35] J. Sotelo et al., "*Char2Wav: End-to-end speech synthesis*," 5th Int. Conf. Learn. Represent. ICLR 2017 - Work. Track Proc., no. 2015, pp. 1–6, 2019.

[36]X. Jin, X. Huang, K. Zhang, and M. Miao, "*On continuous speech recognition of Indian English,"* ACM Int. Conf. Proceeding Ser., 2018, doi: 10.1145/3302425.3302489.

[37] W. Xiong et al., "*The microsoft 2016 conversational speech recognition system*," ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc., no. Lm, pp. 5255–5259, 2017, doi: 10.1109/ICASSP.2017.7953159.

[38] K. Li, H. Xu, Y. Wang, D. Povey, and S.

Khudanpur, "*Recurrent neural network language model adaptation for conversational speech recognition*," Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH, vol. 2018- Septe, pp. 3373–3377, 2018, doi: 10.21437/Interspeech.2018-1413.

[39] H. Soltau, H. Liao, and H. Sak, "*Neural speech recognizer: Acoustic-To-word LSTM model for largevocabulary speech recognition*," Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH, vol. 2017-Augus, pp. 3707–3711, 2017, doi: 10.21437/Interspeech.2017-1566.

[40] Chan, W., Jaitly, N., Le, Q.V. and Vinyals, O.,2015. *Listen, attend and spell*. arXiv preprint arXiv:1508.01211.

[41] P. Sudhakaran, A. K. Yadav, and S. Karamchandani, "*Parasitic sorority of speech processing algorithms with an assortment of statistical toolkits*," J. Phys. Conf. Ser., vol. 1998, no. 1, 2021, doi: 10.1088/1742-6596/1998/1/012024.

[42] MacKenzie, I. Scott, and R. William Soukoreff. "*A character-level error analysis technique for evaluating text entry methods*." In Proceedings of the second Nordic conference on Human-computer interaction, pp. 243-246. 2002.