Speech Recognition using Recurrent Neural Networks

Aditya Amberkar,
MCT's Rajiv Gandhi Institute of Technology,
Mumbai, Maharashtra
adityaamberkar4@gmail.com

Gaurav Deshmukh,
MCT's Rajiv Gandhi Institute of Technology,
Mumbai, Maharashtra
gauravsanjay24@gmail.com

Parikshit Awasarmol,
MCT's Rajiv Gandhi Institute of Technology,
Mumbai, Maharashtra
parikshit024@gmail.com

Piyush Dave
MCT's Rajiv Gandhi Institute of Technology,
Mumbai, Maharashtra
dave.piyush20@gmail.com

***Abstract*— From the past decades there has been rapid change in field of Artificial Intelligence (AI). One of the change is use of AI in field of speech processing .After years of research and development new algorithms were developed some were based on Neural Networks to improve the accuracy of speech recognition system. This paper presents a study of Recurrent Neural Networks and there performance also the use of RNN by famous Speech to Text conversion engines.**

Keywords—Artificial intelligence, Reccurrent neural networks, Speech recognition system , Speech to text .

I.INTRODUCTION

Speech Recognition is also known as Computer Speech recognition which means making the computer understand what we speak. In general program, a computer such that it can hear us and respond us. By 'understand' we mean it would convert our voice into appropriate format for e.g. Text. Thus speech recognition is also called as Speech to Text conversion process .It consists of microphone for humans to speak, recognition of speech software and a computer to perform task. The basic recognition of speech system is shown in fig. 1 [1]

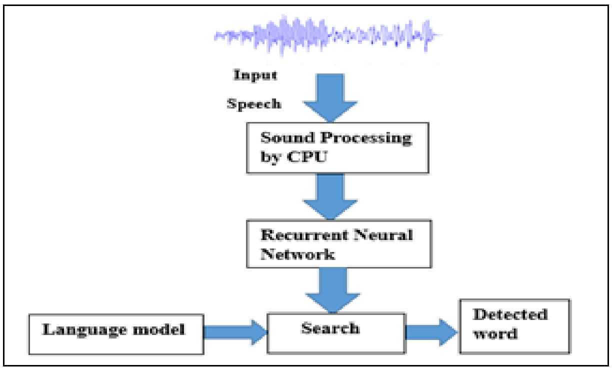


Fig. 1 Speech Recognition system

Speech to text engine

We have to fed sound waves in computer for converting it into text. As sound waves are continuous (analog) signal the first thing to perform is sampling of signal using Nyquist theorem. This sampled signal directly to our neural network but pre-processing of signal is done in order to get better result and accurate predictions of spoken words. Pre-processing is grouping of a large sampled signal into 20-milliseconds small chunks. Pre-processed sampled data which is in digital format is now fed to our Recurrent Neural Network (RNN) which is our main speech recognition model used for prediction. Models used in STT Engine are discussed in further sections.

Sampling and pre-processing of speech

Sampling and pre-processing of data is important step while designing STT Engine .This step decide the performance and time consumption of the engine. Sound waves are one-dimensional. At every moment in time, they have a single value based on the magnitude of the wave. To turn this sound wave into number just record the magnitude of the wave at equally-spaced points. This is called as Sampling shown in fig. 2. [1]

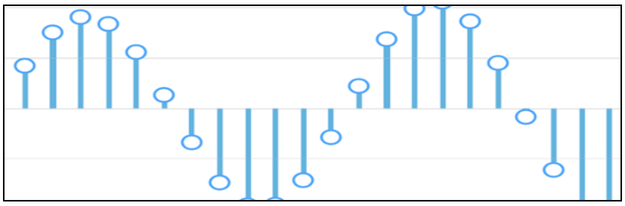
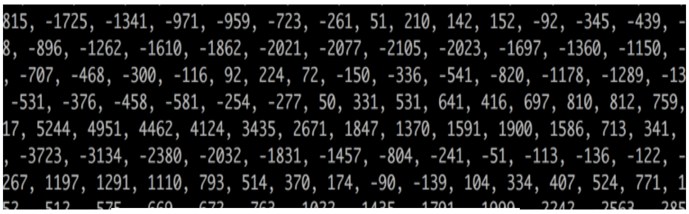


Fig. 2 Sampling of speech signal

Sampling rate is decided using nyquist theorem it is mostly 1/16000th seconds interval [2].Math is used to perfectly reconstruct the original sound wave from the spaced-out samples with sampling frequency equal to or twice more than of highest frequency at which it is recorded. This sampled data directly fed to our recurrent neural network but for ease and better results data is pre-processed before applying to the network. Pre-processing is breaking the sampled data into group of data. Generally grouping the sound wave within some interval of time mostly for 20-25 milliseconds. Sampling and pre-processing together can be termed as conversion of sound into numbers (bits) which shown in fig. 3. [3]



```
815, -1725, -1341, -971, -959, -723, -261, 51, 210, 142, 152, -92, -345, -439, -
8, -896, -1262, -1610, -1862, -2021, -2077, -2105, -2023, -1697, -1360, -1150, -
, -707, -468, -300, -116, 92, 224, 72, -150, -336, -541, -820, -1178, -1289, -13
 -531, -376, -458, -581, -254, -277, 50, 331, 531, 641, 416, 697, 810, 812, 759,
17, 5244, 4951, 4462, 4124, 3435, 2671, 1847, 1370, 1591, 1900, 1586, 713, 341,
, -3723, -3134, -2380, -2032, -1831, -1457, -804, -241, -51, -113, -136, -122, -
267, 1197, 1291, 1110, 793, 514, 370, 174, -90, -139, 104, 334, 407, 524, 771, 1
```

Fig. 3 Sampled and pre-processed data

III.RECURRENT NEURAL NETWORK

Fig. 4 shows speech recognition using RNN. Now audio which is given at input is easy to process, it will be feed into a deep neural network. After feeding small audio chunks of around 20ms to our network it will figure out letter which matches the spoken sound [1,3]

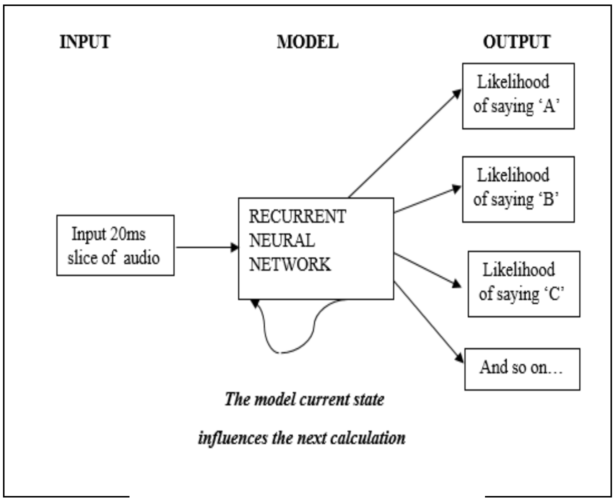


Fig. 4 Speech Recognition Model

RNN is a network which has a memory that decides the future predictions. This is because as it predicts one letter it will affect the likelihood of the upcoming letter which it will predict too. Consider a example, if we have said "MUM" so far, then it's obvious one will say "BAI" next to complete the word "MUMBAI". There is much less probability that one will say something which is unpronounceable "ABC" after saying the word "MUM". Hence having a memory of previous predictions boosts our network to make more accurate predictions going forward. [1,2]

If X=VANILLA-{1 0 0} Y=CHOCOLATE-{0 1 0} Z=BUTTERSCOTCH-{0 0 1}

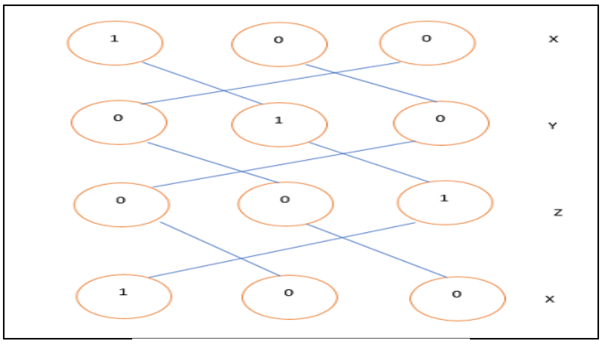


Fig. 5 RNN ice-cream example

A simple recurrent neural network works like the above model. E.g. a person eats vanilla ice-cream on one day ,chocolate on other day and butterscotch on other after that he eats vanilla and chocolate and so on repeats. Here, it means that the input is dependent on the past output and it repeats itself after some time. Hence, the name recurrent neural network. [1,5]

RNNs uses the idea of sequential information. RNN, a neural network that has a memory that influences future predictions Sequential information which is stored in memory of RNNs is used for predictions. Idea to use RNN instead of traditional neural network is in traditional neural network it is assumed that every input & every output are doesn't depends on each other. Hence using traditional neural network is bad idea in speech processing.[3] Prediction of any words in a sentence requires the information about the word which is utilised before i.e. past word which is processed. Having a memory is one of the speciality of RNN that makes it unique than other networks There are various neural networks available among them the Recurrent Neural network [RNN] is used because it is more efficient than the others for speech recognition. [1,4]

Algorithm

Steps involved in RNN algorithm [1] is :

X_t is input at time t , X_t-1 is past input and X_t+1 is the future input (sampled sound)

II. St is the hidden state. It is the hidden memory. St is calculated as: $S_t = f(U*X_t + W*X_t\text{-}1)$.

O_t is output at the step t .For example if we want to predict the next word in a sentence it Would be a vector of probabilities across our vocabulary , O_t=softmax($V*S_t$)

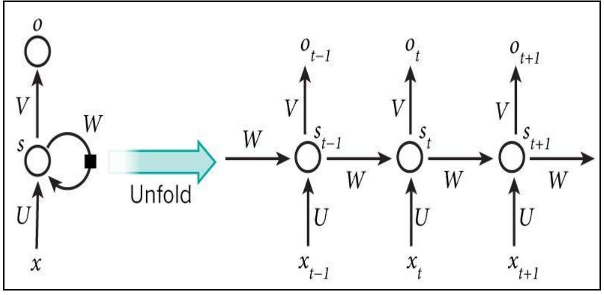


Fig. 6. RNN algorithm

Few things to note here are :
State S_t is the memory of the recurrent neural network that can be hidden. S_t stores the data of what things took place in all the previous or past time steps. Output at step O_t is calculated exclusively based on the memory at time ‘t’. As mentioned above, it’s a little more complicated in practice and practical implementation because S_t normally can’t capture data from too many time steps ago.

- For implementation traditional neural network which are deep uses various parameters at every layer while the same parameters are shared by RNN.
- It uses (U,V,W) parameters as shown by all steps above. This shows us about that we are doing the same task at every single step , by passing various inputs at different step. By this there is a decrease in the number of parameters in all which need to be learned.

Diagram shown above has outputs at each time step, but depending on the task to perform it is not necessary .Consider an example, where we have to predict the sentiment of a sentence here we may only care about the final output, not the sentiment or the output which is given after each word. Same case for inputs too , we don’t need inputs at each time step. Important feature of a RNN is its hidden state, which stores some data about a sequence.

A. *LSTM*

The most commonly used RNN’s are LSTM shown in fig. 7. Long short term memory (LSTM) is building units for multiple layers of RNN.As you know, RNN faces problem of long term dependencies which can be eliminated using a new form of RNN called as LSTM. All RNN are formed consisting of a repeating structure but in case of LSTM the structure varies a bit. It consists of *four* structure unlike RNN which have only one. Main element in LSTM are *cell state* which makes any information to flow through it . Also we can add or remove any information as per requirement using gates. There are three types of gates input gate, output gate and forget gate. Functions of these gates are to protect or control cell state. LSTM network consists of a sigmoid function which has only two values at its output either it will pass everything which it consists at the input or it won’t both the cases are possible. So using a cell state we can control the long term dependencies which was causing problem in case of RNN. Hence LSTM find it’s way to be utilised in newer versions of *speech recognition software*.

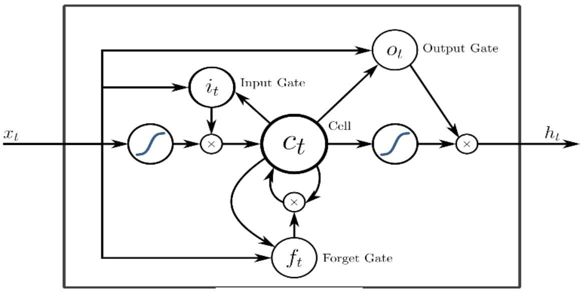


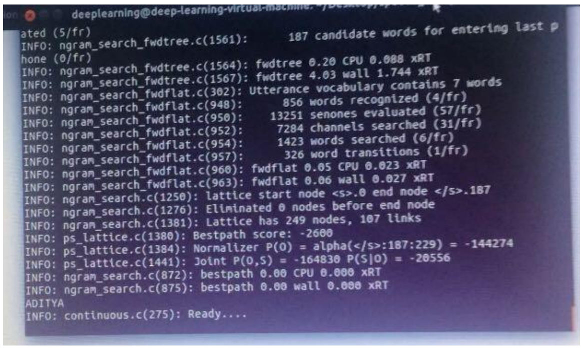
Fig. 7. LSTM

IV.TRAINING A RNN

For training a RNN we use the backpropagation algorithm, but with a approach which differs from a normal one. Reason why approach differ than that of traditional one is at every time steps the parameters used is common throughout the network, the calculated gradient at output step doesn’t depends only on present time step but also o n the computations of the past steps. Consider an example, in which we have to calculate the gradient at *t=5* then you would need to back propagate 4 step and add all the gradients. This method of computation is called as Back propagation through Time (BPTT). One drawback of training a RNN using Back propagation algorithm is long-term dependencies (dependencies on steps which are apart from each other).This problem is called as vanishing or exploding gradient problem. To deal with such type of drawbacks there are some machineries which are utilised and some types of RNNs(like LSTMs) were specifically designed to solve this problems [2]

B. *Various Stt Engines Using RNN*

There are various engines used which are based on RNN’s which uses python ,C, java programming languages to build a Recurrent Neural Network. We have gone through CMU pocket-sphinx, snowboy hot word detection which uses python language and RNN which shows good results also if we increase the database it will perform at it’s best. Unlike Google’s STT and Amazon’s Alexa , CMU pocket-sphinx is offline speech to text conversion engine provided that training of a dataset is done online. Results of testing a speech are shown in fig. 7. CMU online training portal must be given a set of words which we have to train. Training process is same as discussed above in training of RNN. It uses python to build a LSTM network. Also recently launched engine Snowboy-hot word detection works offline but it is limited for detection of one particular hot word.



```
deeplearning@deep-learning-virtual-machine: ~/Desktop/
ated (5/fr)
INFO: ngram_search_fwdtree.c(1561):      187 candidate words for entering last p
hone (0/fr)
INFO: ngram_search_fwdtree.c(1564): fwdtree 0.20 CPU 0.088 xRT
INFO: ngram_search_fwdtree.c(1567): fwdtree 4.03 wall 1.744 xRT
INFO: ngram_search_fwdflat.c(302): Utterance vocabulary contains 7 words
INFO: ngram_search_fwdflat.c(948):       856 words recognized (4/fr)
INFO: ngram_search_fwdflat.c(950):     13251 senones evaluated (57/fr)
INFO: ngram_search_fwdflat.c(952):      7284 channels searched (31/fr)
INFO: ngram_search_fwdflat.c(954):      1423 words searched (6/fr)
INFO: ngram_search_fwdflat.c(957):       326 word transitions (1/fr)
INFO: ngram_search_fwdflat.c(960): fwdflat 0.05 CPU 0.023 xRT
INFO: ngram_search_fwdflat.c(963): fwdflat 0.06 wall 0.027 xRT
INFO: ngram_search.c(1250): lattice start node <s>.0 end node </s>.187
INFO: ngram_search.c(1276): Eliminated 0 nodes before end node
INFO: ngram_search.c(1381): Lattice has 249 nodes, 107 links
INFO: ps_lattice.c(1380): Bestpath score: -2600
INFO: ps_lattice.c(1384): Normalizer P(O) = alpha(</s>:187:229) = -144274
INFO: ps_lattice.c(1441): Joint P(O,S) = -164830 P(S|O) = -20556
INFO: ngram_search.c(872): bestpath 0.00 CPU 0.000 xRT
INFO: ngram_search.c(875): bestpath 0.00 wall 0.000 xRT
ADITYA
INFO: continuous.c(275): Ready....
```

Fig. 8. Implementation using pocket-sphinx STT Engine

From the above figure, words which we want to use is trained using CMU pocketsphinx online tool and then on any linux based or windows based operating system we can implement the code using proper commands by assigning microphone a proper port and trained model of words which we have to test.As described it will get an sound input here it is "A D I T Y A " followed by pre-processing of it and finally feding it to RNN Network which will help to detect the word as shown in the above mentioned figure 7.

V. CONCLUSION & FUTURE USE

RNN is one of the best algorithm used for processing of speech signal and it has scope in emerging voice controlled technologies but training algorithm is again very complex .It shows better results than Multilayer perceptron(MLP).Speech recognition has attracted many scientists and researchers and can be influential to society in emerging technologies. Hope this paper give basic understanding of Speech Processing using Recurrent Network and various STT Engine available which can be used for application development

REFERENCES

[1]. Bhushan C.Kamble,"Speech recognition using artificial neural network" proc of Int'l Journal of Computing, Communications & Instrumentation Engg. (IJCCIE) Vol. 3, Issue 1 2016

[2]. Preeti Saini, Parneet Kaur,"Automatic Speech Recognition: A Review" proc International Journal of Engineering Trends and Technology- Volume4 Issue2-2013

[3]. Yashwanth H, Harish Mahendrakar and Suman Davia, "Automatic Speech recognition Using Audio Visual Cues", IEEE India Annual conference pp. 166-169, 2004

[4]. Urmila Shrawankar, Dr. Vilas Thakare, "Techniques for Feature Extraction in Speech Recognition System: A Comparative Study", (IJCAETS), ISSN 0974-3596,pp 412-418, 2010.

[5]. M.A.Anusuya and S.K.Katti, "Speech Recognition by Machine: A Review", (IJCSIS) International Journal of Computer Science and Information Security, vol. 6, no. 3, pp. 181-205, 2009.