International Journal of Research in Signal Processing, Computing & Communication System Design
5 (1), 2019, 09-14
http://www.publishingindia.com/ijrsp

Deep Bidirectional RNNs Using Gated Recurrent Units & Long Short-Term Memory Units for Building Acoustic Models for Automatic Speech Recognition

Madhuri Jain[1], Nishita Dutta[2], Dnyaneshwari Bhirud[3] and Nikahat Mulla[4]

[1]Sardar Patel Institute of Technology, Andheri, Mumbai, Maharashtra, India. Email: madhurijain97@gmail.com
[2]Sardar Patel Institute of Technology, Andheri, Mumbai, Maharashtra, India. Email: nishita_dutta@outlook.com
[3]Sardar Patel Institute of Technology, Andheri, Mumbai, Maharashtra, India. Email: dobhirud@gmail.com
[4]Sardar Patel Institute of Technology, Andheri, Mumbai, Maharashtra, India. Email: nikahat_kazi@spit.ac.in

Abstract: Deep Neural Networks are gaining popularity to train speech dataset for speech recognition. A lot of work has been done with various neural network models, starting right from conventional convolutional neural networks to deep recurrent neural networks. Research has led us to arrive at the conclusion that bidirectional RNNs are suited for speech recognition. It has been seen that bidirectional RNNs provide greater accuracy as compared to deep RNNs and unidirectional RNNs. Units that are used with bidirectional RNNs are usually Long Short-Term Memory units. They have their own advantages and disadvantages. Gated Recurrent Units can also be used. In this paper we have tried to experiment and compare between deep bidirectional models using GRU units and LSTM units.

Keywords: Acoustic modeling, Automatic speech recognition, Bidirectional RNN, Convolutional neural networks, Deep recurrent neural networks, Gated recurrent unit, Keras, Long Short-Term Memory (LSTM), MFCC, Recurrent neural networks, TimeDistributed dense, TensorFlow, Spectrogram.

I. Introduction

Speech Recognition has gained popularity in the recent years and has been implemented using a wide range of models and neural networks. This paper gives a comprehensive study of the various bidirectional models that we have implemented and experimented on. Audio features including spectrogram features and Mel-Frequency Cepstral Coefficient (MFCC) features were extracted from the speech samples in the dataset and training was done on these features. Focus is on Bidirectional RNNs as they provide maximum accuracy as compared to other unidirectional models. BRNNs can be implemented using both Long Short-Term Memory (LSTM) units or by using Gated Recurrent Units (GRU). We have experimented with various optimizers like SGD, RMSprop, Adam, and learning rates to create a better model for speech recognition than the already existing models.

II. Related Work

In case of Recurrent neural networks, information derived from past inputs is provided with recurrent connections at the hidden layer, while Long Short-Term Memory (LSTM) [1] neural networks are RNNs that can cache data for random amount of time [2] [3]. Lots of models have been proposed before. Better Word error rate up to 16% has been achieved by using RNNs [4]. Instead of using LSTM better performance can be achieved by using deep extensions on LSTM [5].

Max-out unit can be integrated with LSTM cells to achieve better performance. Max-out units have brought significant change in the deep feed-forward neural networks. It is helpful while working on large vocabularies [6]. Recurrent neural network language model are largely used in speech recognition partially as they deal with long distance context than word n-gram models. Feed forward neural networks takes a lot of time for training, by using hierarchical feed forward neural network the training speech can be increased significantly [7].

Bidirectional networks predicts outputs based on both the past and future inputs whereas conventional unidirectional networks predicts output only from past inputs. By applying bidirectional RNNs and LSTM neural networks to language models for speech recognition, further results are improved [8].

Deep Bidirectional Long Short-Term Memory (BLSTM) [9] recurrent neural network acoustic models outperform feedforward neural networks for Automatic Speech Recognition (ASR).The model improves the word error rate over 15% [10].

Deep Long Short-Term Memory RNNs achieve a test set error of 17.7% on the TIMIT phoneme recognition benchmark [11]. After comparing LSTM and GRU recurrent units for polyphonic music modeling and speech signal modeling, it was found out that GRU performs better [12] [13].

III. Tools and Techniques Used

TensorFlow is an open source library that help us compute complex numerical problems with the help of data flow graphs. It is mostly used for deep learning and training neural networks. Our requirements involved easy and fast prototyping, user-friendliness along with modularity and extensibility. For this purpose, Keras which is written in Python and has capabilities pertaining to running with TensorFlow was chosen.

Following Keras dependencies were made use of:

- cuDNN (to run Keras on GPU).
- HDF5 and h5py (required for saving Keras models to disk).
- graphviz and pydot (visualization tools).

TensorFlow runs significantly faster on a GPU as compared with a CPU. All computations were done on an NVIDIA GeForce GTX 1050 with 4GB DDR5 graphics memory.

CUDA® Toolkit 8.0 and all drivers associated with it were needed.

Jupyter Notebooks notebook document format, interactive computing protocol, visualizations and narrative text helped in recording and noting down results obtained at various points during experimentation.

IV. Dataset Used

LibriSpeech is a corpus of approximately 1000 hours of English speech. Its sample rate is a uniform 16kHz and has been prepared by Vassil Panayotov with the assistance of Daniel Povey [14]. The data used for this experiment is derived from audiobooks from the LibriVox project that are available as a LibriSpeech corpus, and has been carefully segmented and aligned.

From this dataset, training was done on 16GB or 100 hours of recording on an approximate of 5000 samples. Testing was done on a subset of this data comprising of 1.96GB of data.

V. Definitions

A. Node

A layer is made by stacking up nodes also referred as neurons. These neurons are what make up different models. Nodes perform two functions:

- Compute summation of linear function *weight*input_feature+bias=z*.
- Using an appropriate activation function, compute the activation of Step 1 for each neuron.

B. Activation Function

Activation functions helps neural networks to learn and understand the input from the node i.e. summation of linear function. These activation functions f(x) introduce non linearity which helps to learn from complicated and huge input. These functions must be differentiable so that the loss can be computed during back propagation. In this paper, we have used the following linear function:

(i) ReLU (Rectified Linear Unit) Activation Function

This can only be used in the hidden layers and converges six times faster than tanh activation function. ReLU helps to avoid and rectify vanishing gradients problem. The mathematical formula of the function is:

$$R(x) = \max(0, x), \quad \text{if } x < 0$$
$$= x \quad \text{if } x >= 0 \tag{1}$$

(ii) Softmax Activation Function

This activation function is used to compute the probability of output when there are multiple classifications. It is a generalization of logistic regression. In this paper, these multiple classifications includes letters from a to z and special characters.

C. GRU (Gated Recurrent Unit)

GRU is one of LSTMs variation, it is similar to LSTM but requires less parameter comparatively. It addresses the vanishing gradient problem which occurs in the normal recurrent networks. GRU uses only two gates, update gate and reset gate. Certain information is needed to be passed to the output, that is done by these two vectors. They are trained so that they can sustain information for a long time. They don't lose data or remove certain data for any kind of prediction.

D. LSTM (Long Short-Term Memory)

The normal recurrent neural network has a disadvantage of long short-term dependency. LSTM is explicitly designed to avoid the long short-term dependency They have memory blocks called as cells. They are used to sustain information for the long period of time. LSTM transfers two states to the next cell, the cell state and the hidden state. LSTM uses three gates viz output gate, input gate, and forget gate, These three mechanisms are used to perform manipulation on the memory in LSTM [15] [16].

E. Spectrogram

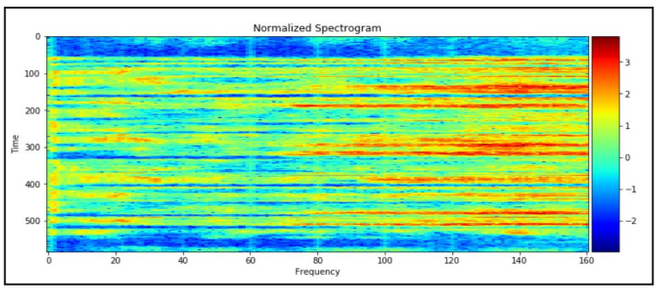

Fig. 1: Normalized Spectrogram

Spectrogram, is a sound feature, used to understand timbre. It gives a visual representation of the frequency of sound. It gives us details of individual vocal folds, harmonics and acoustic patterns. Spectrogram features follow linear frequency scaling. Hence, all the frequency bins are equal hertz apart from each other.

F. Mel-Frequency Cepstral Coefficient (MFCC)

MFCCs are features used in speech recognition. It is a set of coefficients used to represent short term power spectrum of sound. Due to the absence of resistance to noise, MFCC is not considered robust. A MFCC feature vector consists of 13 cepstral coefficients, 13 first and second order derivatives. Since MFCC features are dynamic and mimic human perception, it is preferred over Spectrogram feature vectors.

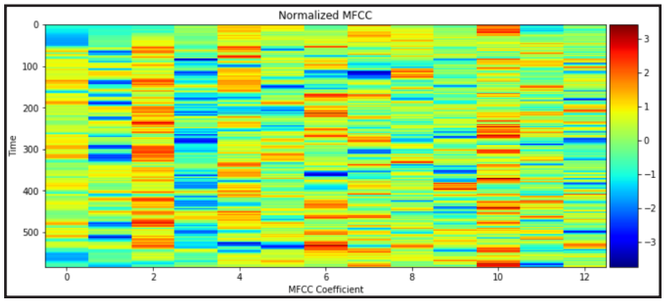

Fig. 2: Normalized MFCC

The MFCC features are calculated through the following steps:

- Fast Fourier transform on the received speech.
- Mel Scale Filtering- It computes the power of each frequency band since humans use power than frequency for signal processing.
- Logarithm- This is used to replicate the human perception of loudness.
- Discrete Cosine Transform.
- Derivatives of these Cepstral Coefficients obtained from DCT.

A MFCC feature vector consists of 13 cepstral coefficients, 13 first and second order derivatives. Since MFCC features are dynamic and mimic human perception, it is preferred over Spectrogram feature vectors.

G. TimeDistributed Dense (TDD)

TDD is a wrapper to normalize the activations of the previous input layer before feeding it to the next hidden layer. TimeDistributed performed on the output layer is referred as flattening since the output is usually a linear vector. Softmax activation function is applied to this layer which is used to predict when multiple output classifications (29 classifications in this paper) are involved.

H. Optimizers

(i) *SGD:* Stochastic gradient descent or incremental gradient descent, is an approximation of the gradient descent optimization and iterative method for minimizing an objective function that is a sum of differentiable functions. Learning rate plays an important role in stochastic gradient descent.

(ii) *RMSprop:* RMSprop allows you to have higher learning rates. It divides the learning rate by an exponentially decaying average of square gradients. It ensures that the denominator is not very close to zero.

(iii) *Adam:* Adam (Adaptive Moment Estimation) is an optimisation algorithm that has evolved from the classical SGD algorithm and is an update to RMSProp optimizer. It is computationally efficient and requires little memory. Adam makes use of algorithms like AdaGrad and RMSProp. An alternative to Adam could be SGD + Nestorov Momentum [17].

I. Connectionist Temporal Classification Loss

It is useful for performing supervised learning on sequence data, without needing an alignment between input data and labels. In speech audio, there are multiple time slices that can correspond to a single phoneme. Prior to prediction we

do not know the alignment of the observed sequence and its true transcript. Hence, with the help of the CTC we predict a probability distribution at each time step. This probability distribution is then used to fit our data and the true labels of the audio samples. These CTC scores are used to update weights in the next iteration through back propagation mechanism of our designed RNN model [18].

J. Word Recognition Rate

Word Recognition Rate (WRR) is calculated in percentage which indicates how close the predicted label is to that of the true label. It calculates how accurately were the transcriptions predicted. Word Error Rate (WER) is the Levenshtein distance for words, and WRR is (100% - WER). To measure the accuracy and performance of our models, we calculated the average of WRR over all the samples in the testing set.

VI. Model Architectures

A. Bidirectional RNN (GRU)+TDD

Conventional RNNs have a disadvantage that they can only use data from the previous context. In speech recognition, an entire word is transcribed in a single go, therefore, it is beneficial if we consider both future as well as previous context. Bidirectional RNNs (BRNNs) can retrieve this future input by processing the data in both directions with two separate hidden layers [2].

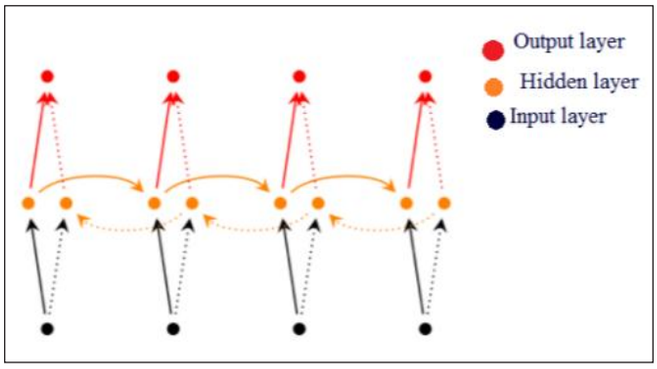

Fig. 3: Basic Bidirectional Model

The model was trained with GRU using MFCC and Spectrogram features. Each recurrent layer comprised of 200 neurons and as a result the complete bidirectional layer comprised of 400 neurons. The results obtained for MFCC features were comparatively better than that obtained with Spectrogram.

B. Bidirectional RNN (LSTM)+TDD

This model was trained with LSTM using MFCC and Spectrogram features. The model was trained with one bidirectional RNN each of 400 neurons having 200 neurons in the forward and 200 neurons in the backward layer.

VII. Input and Output

Time sequence of audio features, i.e. either MFCC features or spectrogram features are fed as input to the network. MFCC features consists of 13 dimensions while spectrogram features are 161 dimensional. Note here that chances of over fitting our training set has a higher probability with spectrogram features.

Our input vectors, be it MFCC or spectrogram are normalized so that convergence can be achieved faster.

The output consists of 29 entries, where each entry corresponds to the probability of the i-th letter of the English alphabet. The remaining three letters are the comma (,), space (“ ”) and the apostrophe (’). The probability represents the probability with which the letter might have been spoken in the audio at that particular time sequence.

VIII. Comparison of Models

A. Bidirectional Model Comparison

We trained BRNNs using GRU and LSTM with Spectrogram and MFCC features. The variation of loss obtained on the training set and the validation set are given in Fig. 1. The results clearly indicate that BRNN with LSTM using MFCC feature gives the lowest loss and highest accuracy compared to all the experimented models stated in this paper. Both GRU and LSTM models show similar behavior at the end of 20 epochs and converge to a loss of around 114.

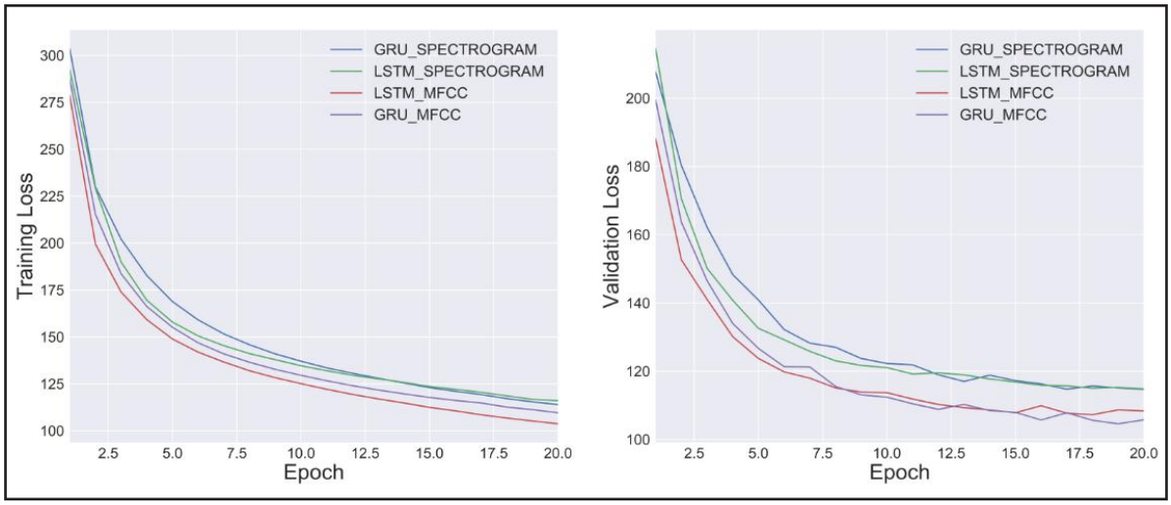

Fig. 4: Loss on Training Set and Validation Set

B. Results

The results of the trained models mentioned earlier are tabulated where the efficiency and preferred model is decided using parameters like loss on the training set, and the accuracy determined using WRR. These models were trained with 20 epochs each comprising of 260 input samples. WRR was found using fuzzywuzzy library. The details of the results of these models are given in Table I. Models that use spectrogram features both start off with a high initial loss and come down to a loss of 113 and 116 for GRU and LSTM respectively. The LSTM spectrogram model's loss values start to stagnate at the sixteenth epoch at a constant value of around 115. It shows little improvement in the further epochs. The GRU MFCC model fairs better and shows constant improvement over 20 epochs. LSTM MFCC model performs the best amongst all models and shows a constant rate of decrease in loss values. Even the initial loss values are lower.

TABLE I: LOSS AND ACCURACY FOR BRNN MODELS

BRNN + TDD Models: Loss and Accuracy						
Cell Type	Feature Used	Initial Validation Loss	Final Validation Loss	Initial Training Loss	Final Training Loss	Word Recognition Rate (%)
GRU	Spectrogram	207.68	114.67	303.57	113	45.79
LSTM	Spectrogram	214.45	114.83	292.47	116.05	48.077
GRU	MFCC	199.43	105.75	287.57	109.60	50.22
LSTM	MFCC	188.10	108.36	278.66	103.72	52.464

IX. CONCLUSION

A comparison of Bidirectional models was done against the type of cells i.e. GRU or LSTM using different features, Spectrogram and MFCC. The best model amongst these was found to be LSTM with MFCC feature set, with a word recognition rate of 52.464%. Models that used MFCC features were found to perform better than those that used spectrogram features.

Deeper models with greater number of hidden layers will significantly affect the recognition rate and performance of the models.

Models could be trained on larger dataset and on more epochs. Language and grammar models can be incorporated to improve accuracy of the models. CNNs can be trained in combination with BRNNs to improve the accuracy. Max pooling can be done to improve CNNs. By training the model in this way, it can be used to transcribe videos and generate subtitle files, recognize non-speech audio events.

REFERENCES

[1] S. Hochreiter, and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 1997.

[2] M. Schuster, and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673-2681, 1997.

[3] T. Mikolov, A. Deoras, D. Povey, L. Burget, and J. Cernocky, "Strategies for training large scale neural network language models," in *2011 IEEE Workshop on Automatic Speech Recognition & Understanding (ASRU)*, 2011.

[4] A.-R. Mohamed, F. Seide, D. Yu, J. Droppo, A. Stoicke, G. Zweig, G. Penn, "Deep bi-directional recurrent networks over spectral windows," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2015.

[5] X. Li, and X. Wu, "Constructing long short-term memory based deep recurrent neural networks for large vocabulary speech recognition," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* 2015.

[6] X. Li, and X. Wu, "Improving long short-term memory networks using maxout units for large vocabulary speech recognition," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015.

[7] H.-K. J. Kuo, E. Arisoy, A. Emami, and P. Vozila, "Large scale hierarchical neural network language models," in *Proceedings of Interspeech*, Portland, Oregon, USA, 2012.

[8] E. Arisoy, A. Sethy, B. Ramabhadran, and S. Chen, "Bidirectional recurrent neural network language models for automatic speech recognition," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* 2015.

[9] A. Graves, N. Jaitly, and A.-R. Mohamed, "Hybrid speech recognition with deep bidirectional LSTM," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 273-278, 2013.

[10] A. Zeyer, P. Doetsch, P. Voigtlaender, R. Schluter, and H. Ney, "A comprehensive study of deep bidirectional LSTM RNNs for acoustic modeling in speech recognition," in *ICASSP 2017 Conference*, IEEE, 2017.

[11] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* 2013.

[12] J. Chung, C. Gulcehre, K. H. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling,". Available at arXiv:1412.3555 [cs.NE]

[13] Z. Wu, and S. King, "Investigating gated recurrent networks for speech synthesis," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* 2016.

[14] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* 2015.

[15] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition,". Available at arXiv:1402.1128 [cs.NE]

[16] M. Sundermeyer, R. Schluter, and H. Neym, "LSTM neural networks for language modeling," in *Proceedings of Interspeech*, 2012.

[17] D. P. Kingma, and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference for Learning Representations (ICLR),* 2015. Available at arXiv:1412.6980 [cs.LG]

[18] A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, PA, 2006.