Multimedia Tools and Applications (2022) 81:17169–17184
https://doi.org/10.1007/s11042-022-12723-4

Acoustic model with hybrid Deep Bidirectional Single Gated Unit (DBSGU) for low resource speech recognition

S. Girirajan [1] · A. Pandian [1]

Received: 26 December 2020 / Revised: 6 August 2021 / Accepted: 21 February 2022 /
Published online: 5 March 2022

Abstract
Nowadays Long Short-Term Memory RNNs (LSTM RNNs) are widely used in Automatic Speech Recognition (ASR) and achieved excellent result in the problem of vanishing gradients. Bidirectional LSTM (BLSTM) will run the inputs in two ways ,both past as well as in future that shows good performance. However implementation of BLSTM is quite difficult because of its high computational requirements and also the problem of vanishing gradients still persist, when we have multiple layer of LSTM. The extensive size of LSTM systems makes them powerless in over fitting issues. The Gated Recurrent Unit (GRU) is the latest generation recurrent neural networks with two gates. The update gate acts similar to forget and input gates of LSTM's and reset gate responsible to decide how much previous data you should remember. GRU avoids over fitting and also training the GRU is faster compared to LSTM, since size of the GRU network is small. The proposed work is in two- fold architecture. First stage, we tend to reduce the gates in GRU by combining the reset and update gate together to form a Single Gated Unit (SGU). SGU takes half of the parameter compared with LSTM and one third of parameter compared with GRU. It increases the training speed of SGU. Second stage, SGU is combined with Deep Bidirectional design (DBSGU) to build a hybrid acoustic model that takes less number of parameters and increases the learning capability. The proposed model is compared with similarities and differences between Deep Bidirectional GRU (DBGRU) and Deep Bidirectional LSTM (DBLSTM) and found that 2 to 4% decrease in Word Error Rate (WER).The Learning rate of the is increased by 30% The entire work has been evaluated on Crowd Sourced high-quality Multi-Speaker speech (CSMS) data set.

✉ S. Girirajan
girirajans.cse@gmail.com

A. Pandian
pandiana@srmist.edu.in

[1] Department of Computer Science and Engineering, School of Computing, SRM Institute of Science and Technology, Chennai, India

Keywords Automatic Speech Recognition (ASR) · Gated recurrent unit (GRU) · Deep bidirectional · Deep bidirectional single gated unit (DBSGU) · Word Error Rate (WER) · Long short-term memory (LSTM)

1 Introduction

Automatic Speech Recognition (ASR) system is used to recognize the words in the given speech signal and convert them into corresponding text transcript. In recent year ASR system is widely used to control the electronic gadgets like Amazon Alexa, Google Assistant etc. in the form of personal assistant. Initially Hidden Markov Model (HMM) is used to find the phone from the speech signal. In the early stage HMM is used to recognize the words from the given speech that takes the past and future data to predict the present state. It is a statistical model where the modeling system is supposed to be a Markov process with unknown parameters; the challenge is to work out the hidden parameters of the observed data. In the process of recognition, several variants of HMM were used which belongs to either discrete or continuous density model [3] .The HMM is a memory less model due to that the process will not have previous state so each observation is treated individually. Consequently created sentences by a HMM are conflicting. Recurrent Neural Network (RNN) overcome these issues by generating each character based on the reference of past history of characters generated [18] .Particularly LSTM RNNs, are successful system for time series data like speech recognition. Performance of Deep LSTM model over the large vocabulary continuous speech recognition is excellent, due to their noteworthy learning capacity [14]. LSTM is totally depends on the network that contains three gates such as input, forget and output gate to control the memory cells. Implementing LSTM is highly difficult due to its complex structure and computational complexity is also increased [8, 11, 24].

Bidirectional LSTM (BLSTM) is an algorithm that processes the sequential data. The information from both forward and backward states was passed simultaneously to the output layer. BLSTM takes up the in build memory that stores the previous processed data into it. Due to this it avoids the reiteration of process to recognize the phone again in the different context. Along with advantages of LSTM, bidirectional architecture will process the information in both forward and backward to minimize the long range dependency [23]. To overcome vanishing gradients problem Gated Recurrent Unit (GRU) is used. Forget gate is also available in GRU as like LSTM. GRU contains less parameter when compare to LSTM. GRU is more comfortable for smaller dataset and it also produce excellent result in speech signal modeling when compare with LSTM. The LSTM encompass 3 gates particularly input gate, forget gate and output gate. Input gate directs the amount of the new cell state to keep, the forget gate controls the measure of the present memory to discard, and the output gate deals with to measure of the cell state. It should be introduced to the accompanying layers of the framework. The GRU works utilizing an update and reset gates. To overlook the previous state, reset gate is placed between the previous and future state activation, and the update gate chooses the amount of the information initiation to use in refreshing the cell state [20].

In spite of the fact that LSTM models have accomplished amazing outcomes for large vocabulary continuous speech recognition, they still battle when connected to specific task, for example, preparing for low-resource dialects. It is hard to implement Conventional LSTM (CLSTM) and Bidirectional LSTM (BLSTM) over complex training mechanism and also the problem of vanishing gradient over multiple layers is a major issue. We would like to resolve

these deficiencies by utilizing additional gating mechanisms that reduces the complexity in training mechanism and also overcome the temporal dependencies [10]. In the proposed work, SGU is introduced to reduce the complexity further by maintaining the accuracy. Data flow in GRU is controlled by two gates. These two gates must contain correlation and redundancy. Since input and previous hidden state information present in current hidden state are controlled by reset and update gate. Correlation by cross-correlation is proved by Micro et al. [20] by using this reset and update gate shares the same value. Based on this, 2 gates in GRU is further reduced to single gate by coupling reset and update gate to form a single gated unit (forget gate). Further Deep bidirectional design combined with SGU to make a hybrid acoustic model that reduces the time taken to train the model with less number of parameters and also maintain the accuracy comparatively. This proposed model performed well in vanishing gradient issue. Deep SGU build by placing multiple SGU layers in the form of stack [7]. Similarly bidirectional design is used to process data in both forward and backward direction with separate parameter that helps in creating both previous and future context. The proposed architecture reduces the 20% of training time per epoch.

The remaining part of the paper is arranged as follows. Section 2 details with Background Section 3 Proposed Work DBSGU are explained Section 4 describes Experimental Setup Section 5 Experimental Result Segment 6 Conclusion.

2 Background

2.1 Recurrent Neural Network (RNN)

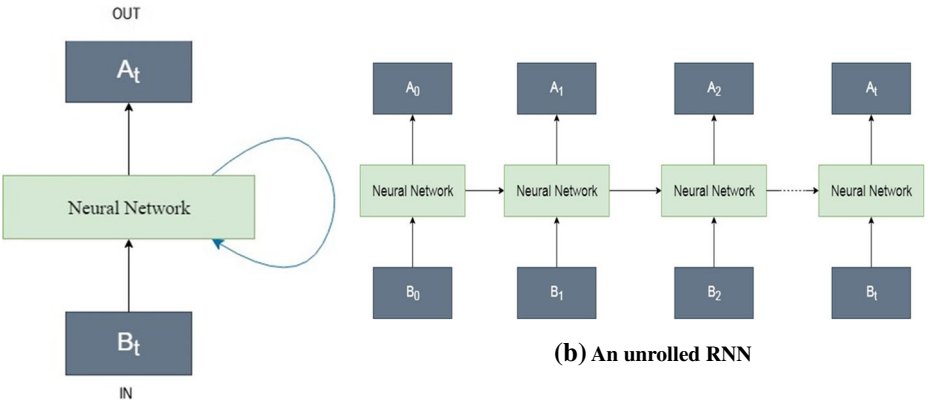
Problems with sequence prediction have been around for quite a while. In data science industry they are considered as one of the most difficult issue to understand. These incorporate a wide scope of issues like stock market prediction, predicting a next word from a speech signal in machine translation.

LSTM plays a major role in sequence prediction issue. LSTM perform well when compared to traditional RNN in various aspects. The inbuild memory that are available in LSTM, avoids the repetition of process for the same phone recognition that occurs in different part of speech signal. In the sequence data prediction, context between the past and present information plays the important role but traditional RNN maintains the context due to that learning rate of the model is gradually increased [12].

LSTM has various advantages over RNN and conventional feed-forward neural network. In sequence prediction problem past history of data plays a major role but conventional feed-forward neural network consist of individual data in all test cases. RNN accomplished this dependency of time [12]. The network design structure of RNN and unrolled RNN is shown in Fig. 1(a) and (b).

2.2 Long-term dependencies problem

RNNs are the idea that they are most likely to associate past data with the present undertaking. Sometimes, the recent information is carried out from the present task. In most of the situation there will be a extensive gap required between the data [2]. Unfortunately, as this gap develops, RNN ends up helpless to find out how to associate data.



(a) Recurrent Neural Networks

(b) An unrolled RNN

Fig. 1 (**a**) Recurrent Neural Networks. (**b**) An unrolled RNN. A_t – output B_t– Input

2.3 LSTM networks

All RNNs have a chain of reiterating neural network modules. Likewise, LSTM has an equivalent structure, yet the repeating module has a substitute structure. There are four interfacing in LSTM which works excellently than a single neural system layer shown in Fig. 2. Equation (1) is for forget gate decides which information need to be discard from the cell state. Equation (2) is for input gate that decides what new information that needs to be stored in the cell state. Equation (5) is for output gate, that generates LSTM final output layer for the timestamp't' by providing the activation function. Equations (3), (4) and (6) denotes the cell state, candidate cell state and the final output.

Steps involved in LSTM:

Step 1: Sigmoid layer in forgot gate chooses what data the cell state has to discard.

$$f_t = \sigma(W_f \cdot [A_{t-1}, B_t] + x_f) \tag{1}$$

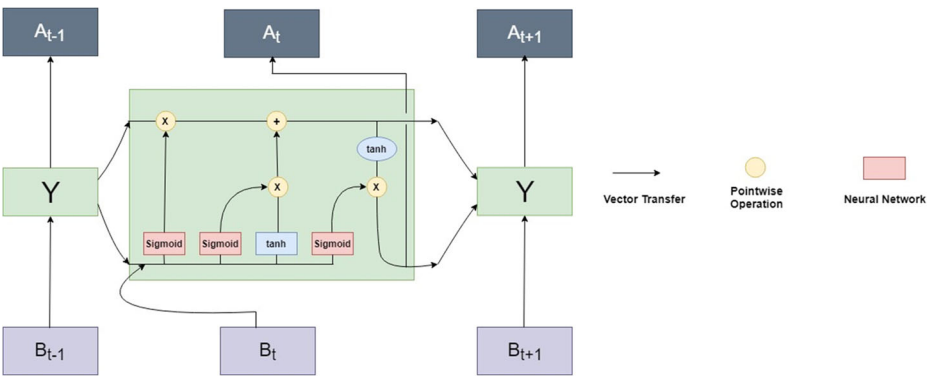


Fig. 2 LSTM with four interacting layers

Step 2: Information get's fed into the input layer. This layer decides what data from the candidate should be added to the new cell state [21].

$$i_t = \sigma(W_i \cdot [A_{t-1}, B_t] + x_i) \tag{2}$$

$$\mathcal{C}_t = tanh(W_C \cdot [A_{(t-1)}, B_t] + x_C) \tag{3}$$

$$C_t = f_t * C_{t-1} + i_t * \mathcal{C}_t \tag{4}$$

$$O_t = \sigma(W_0 * (A_{t-1}, B_t) + x_0) \tag{5}$$

$$A_t = O_t * \tanh(\mathrm{C_t}) \tag{6}$$

A_{t-1} represents output of the previous cell (or) LSTM, B_t input at that particular time, C_{t-1}, C_t,$\mathcal{C}_t$ represents old cell state, new cell state and new candidate value, f_t forget gate state, O_t output gate, i_t input gate, σ sigmoid function, x bias for the respective gate, W weight for the respective gate

2.4 Bidirectional Long Short-Term Memory (BLSTM)

One shortcoming of ordinary RNNs is that they can just utilize the previous context. In speech recognition, where whole utterance are translated without any delay. Bidirectional RNNs (BRNNs) [21] do this by handling information in the two distinct hidden layers, outcome of such layer are then given to a comparable output layer. We should take output [-1, :, :hidden size] for normal RNN $\overrightarrow{A}$ and output[0, :, hidden_size:] for reverse RNN $\overleftarrow{A}$, and then combine those to output and feed the result to the subsequent dense neural network y [1]. BLSTM can handle two separate information in both forward and backward direction with two individual hidden layers [16]. Long range dependency data can be handled by using BLSTM along with the feedback for the next layer. Equations (7) and (8) carries the information in forward and backward direction respectively. Equation (9) is the output layer that get combined data as the input from $\overrightarrow{A_t}$ and $\overleftarrow{A}$ (Fig. 3) [6].

$$\overrightarrow{A_t} = H(W_{B\overrightarrow{A}} B_t + W_{\overrightarrow{A}\overrightarrow{A}} \overrightarrow{A}_{t-1} + x_{\overrightarrow{A}}) \tag{7}$$

$$\overleftarrow{A} = \mathrm{H}\left(\mathrm{W}_{\mathrm{B}\overleftarrow{\mathrm{A}}}\mathrm{B_t} + \mathrm{W}_{\leftarrow\leftarrow\mathrm{AA}}\overleftarrow{\mathrm{A}}_{\mathrm{t-1}} + \mathrm{x}_{\overleftarrow{\mathrm{A}}}\right) \tag{8}$$

$$y_t = W_{\overrightarrow{A}y} \overrightarrow{A}_t + W_{\overleftarrow{A}y} \overleftarrow{A}_t + x_y \tag{9}$$

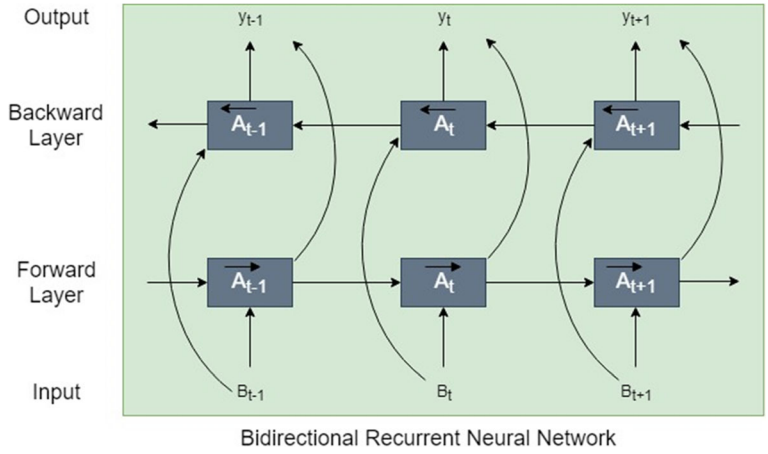


Fig. 3 Bidirectional Recurrent Neural Network (BLSTM)

Combining BRNN with LSTM provides bidirectional LSTM [6] long-range contexts can be accessed in both directions forward as well as backward.

2.5 Gated Recurrent Unit (GRU)

Introduced by Cho, et al. in 2014, GRU (Gated Recurrent Unit) aims to solve the vanishing gradient problem which comes with a standard recurrent neural network. Like the LSTM unit, however without having separate memory cells, the GRU has gating units that regulate the stream of data inside the unit [4, 15] (Fig. 4) [15].

Where the activation of the memory cell A_t at the time t could be a linear interpolation of the previous initiation $A_{t-1}$and the activation candidate A'_t at the time t, r_t is the reset gate and z_t is the update gate. The W terms indicate matrices of weight [5].

Using Eq. (10) update gate is calculated for the time step 't'. update gate decides how much of information from the past need to be forwarded to future.

$$z_t = \sigma(\mathrm{W}^{(z)} B_t + U^{(z)} A_{t-1}) \tag{10}$$

Reset gate decides how much of past information need to be discarded, Eq. (11) performs the reset gate operation.

$$r_t = \sigma(\mathrm{W}^{(r)} B_t + U^{(r)} A_{t-1}) \tag{11}$$

New memory cell is introduced to store relevant information from the past as shown in Eq. (12).

$$A'_t = tanh(\mathrm{WB_t} + \mathrm{r_t} \odot \mathrm{UA_{t-1}}) \tag{12}$$

Equation (13) shows the final memory of the current time step

$$A_t = z_t \odot A_{t-1} + (1 - \mathrm{z_t}) \odot A'_t \tag{13}$$

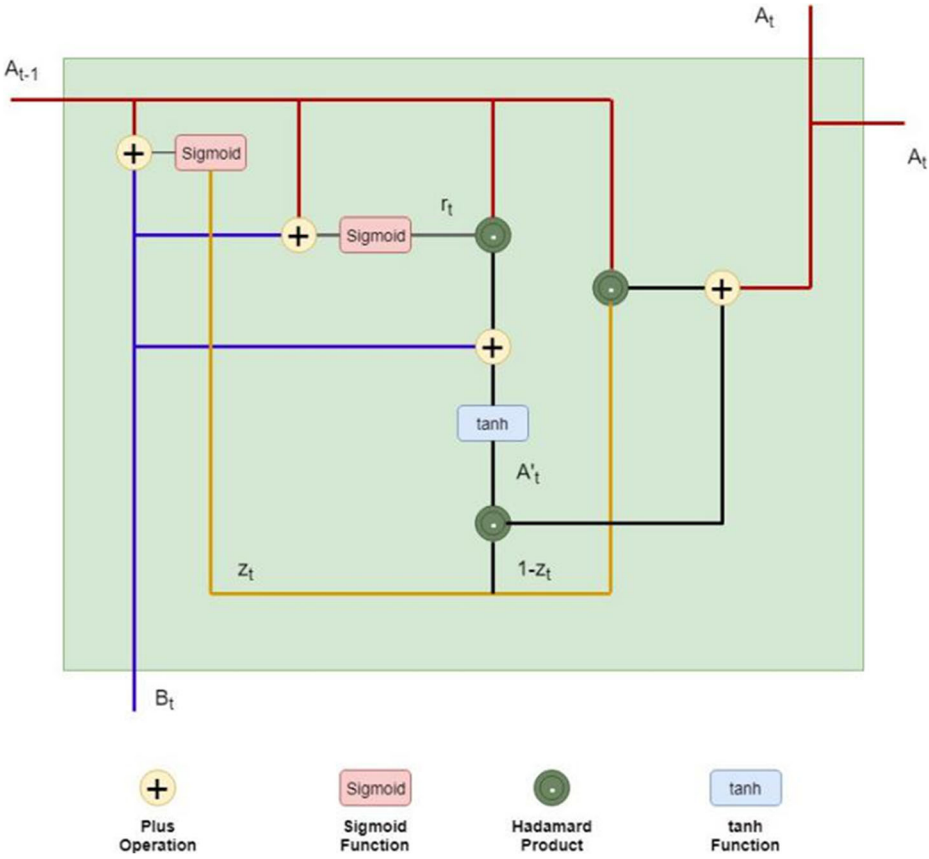


Fig. 4 Gated Recurrent Unit (GRU)

2.6 Single Gated Unit (SGU)

The Single Gated Unit (SGU) is proposed [25] to minimize the gates. There are two gates available in GRU which is further reduced to single gate with the help of SGU. Update gate in GRU is shared with reset gate successfully to form a SGU this can be calculated with help of Eq. (14).

The forget gate is basic and its inclinations bf must be instated to huge qualities; the information input gate is significant, yet the output gate is insignificant; GRU and LSTM have comparable execution [11] .The output and forget gates are basic, and numerous variants of LSTM (fundamentally simplified LSTM variants) act correspondingly to LSTM [8] .Gated units work extremely well to basic units with no gates; GRU and LSTM has practically identical precision with a similar number of parameters [4] (Fig. 5) [25].

Update gate in SGU shown in Eq. (14) also carried out in same way like Eq. (11) used in GRU. That couples the two gate update and reset gate to form a single forget gate.

$$f_t^i = \sigma(U_f A_{t-1} + W_f B_t + x_f)^j \tag{14}$$

where the superscript indicates the gate vector's *j-th* element. Compared to the GRU, the activation status update equations and the j-th element candidate activation becomes:

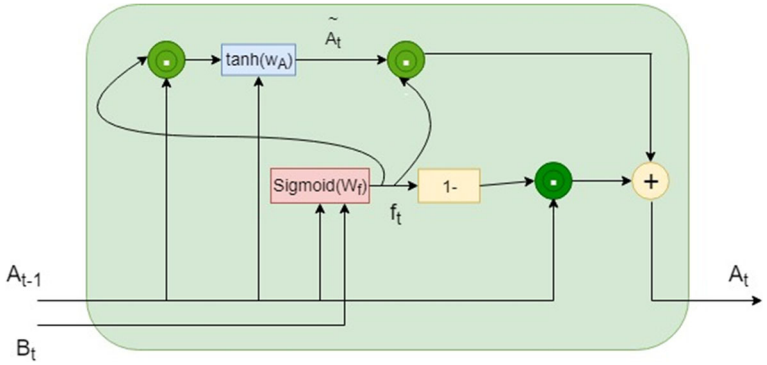


Fig. 5 Single Gated Unit (SGU)

$$A_t^j = ((1 - f_t) \odot A_{t-1} + f_t \odot \hat{A}_t)^j \tag{15}$$

$$\hat{A}_t^j = \tanh(\mathrm{U}(\mathrm{f_t} \odot \mathrm{A_{t-1}}) + \mathrm{WB_t} + \mathrm{x})^{\mathrm{j}} \tag{16}$$

This constitutes a rise of roughly two-fold (adaptive) parameters compared to the RNN. When compared to GRU, SGU has reduced the number of parameters due to that SGU will be trained faster [25].

3 Proposed work : Deep Bidirectional Single Gated Unit (DBSGU) - RNN model description

A vital component of the ongoing accomplishment is utilization of deep bidirectional system that can develop continuously larger amount of acoustic data. RNN Hidden layers are placed over one another and arranged typically in neat order to form a Deep RNNs; input for each layer in the architecture is gathered from output of previous layer, as appeared in Fig. 6. Data is processed in both forward and backward in bidirectional RNN with two separate hidden layers and then processed information is given to same output layer. By repeatedly executing the hidden vector sequences A^n, time t begins at 1 and terminated at T similarly n starts from 1 and ends at N, expecting the equivalent number of hidden layer is utilized well in the architecture.

$$A_t^n = H(W_{A^{n-1}A^n} A_t^{n-1} + W_{A^nA^n} A_{t-1}^n + x_A^n) \tag{17}$$

Where $\mathrm{A^0} = \mathrm{B}$ is specified. The network produces $\mathrm{y_t}$ are

$$y_t = W_{A^N y} A_t^N + x_y \tag{18}$$

Proposed model combines the Deep Bidirectional architecture with SGU to form DBSGU. Figure 7 demonstrates the general structure of the proposed framework.

In bidirectional architecture hidden layer contains one forward SGU layer and one backward SGU layer. Since it is difficult to conclude whether forward or reverse propagation will progressively fit well. We have designed a model in such a way it will act in forward as well as

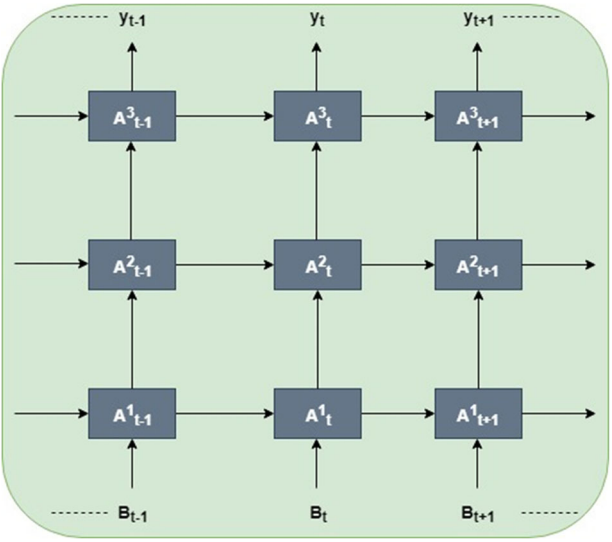


Fig. 6 BRNN

reverse direction. In Bidirectional design not all layers dependent upon its previous layer. Each layer can transmit the information to more than one layer.

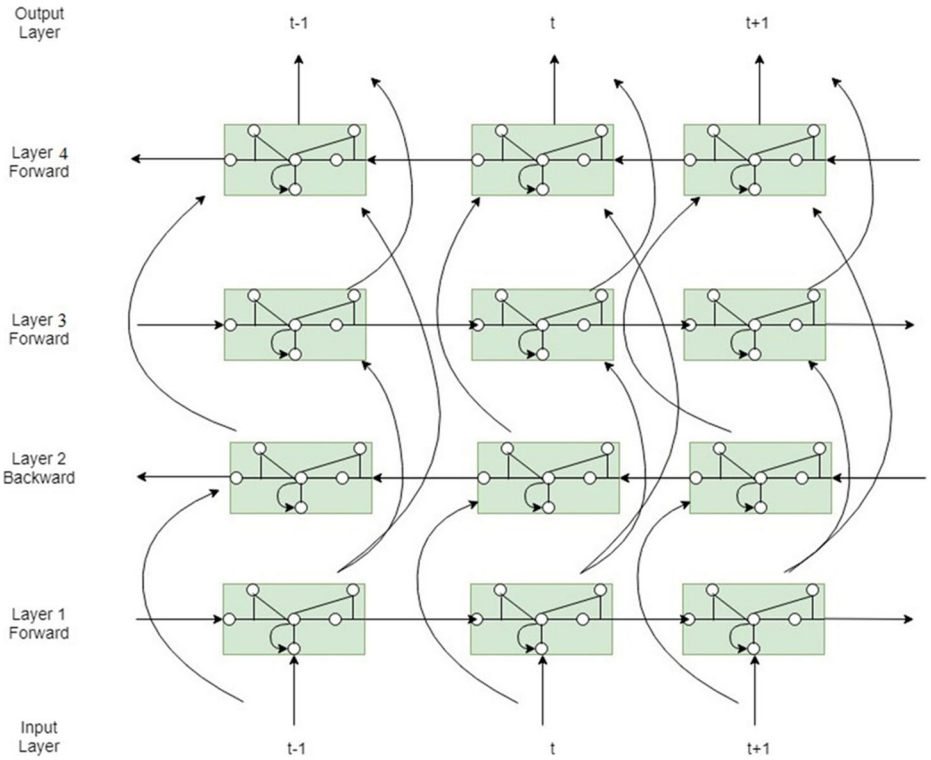


Fig. 7 Deep Bidirectional Single Gated Unit

In proposed SGU, we have used only minimum number of gates when compared to other gated unit. We have set only one forget gate that combines the functionality of both reset and forget gate. It is denoted as

$$r_t = f_t, \forall t \tag{19}$$

f_t indicates that we have used one gate that is forget gate. In Eq. (19), f is used instead of z that denotes the single gate. This single gate is considered as the forget gate. In proposed method first we generate the forget gate f_t then we calculate the product of each element between $1 - f_t$ and A_{t-1} and make it as a new hidden state A_t. A_t is produced by combining the forget gate with x_t.

High performance is achieved by having gated unit of RNN architecture. In overall evaluation more importance is given to forget gate. By reducing the gates, accuracy is maintained with reduced complexity.

From the Eqs. 14 to 16 and Fig. 5 it is clear that SGU is much simplified when compared to LSTM and Gated Recurrent Unit. In Table 1 we can see the different number of parameters required for LSTM, GRU and SGU. You can see that SGU required only minimum number of parameters that makes it easier to process. With less number of parameters we can avoid the gradient vanishing problem. Since we need only a least number of factors to tune.

4 Experimental setup

4.1 Corpus details

Table 2 shown below describes the various properties of the dataset. Both male and female volunteers are used to create a corpus. Source to generate the dataset is collected from Wikipedia. Volunteers are in the age limit of 21 to 35. The crowd sourced high-quality multi-speaker speech data set contains speech corpus for various languages like Tamil, Telugu, Malayalam, Gujarati and soon on [9]. In this proposed work we used Tamil Language alone for training and testing purpose.

Along with the .wav file this data set also contains the text transcript for corresponding audio speech file. The data set consists of 153 h of male and 7440 min of female training data set together with text transcription for Tamil, Telugu and Malayalam languages. For experimental purpose we have used windows 10 Operating System with NVIDIA GTX 1650 GPU is used. The entire work is implemented using python 3.7.

4.2 Data preprocessing

Automatic Speech Recognition converts a raw audio file into character sequences; the preprocessing stage converts a raw audio file into feature vectors of several frames. We should first split each audio file into 32ms Hamming windows with an overlap of 12ms, and then

Table 1 Set of parameters

Sl.No	Methodology	Set of parameters
1.	LSTM	4
2.	GRU	3
3.	SGU	2

Table 2 Dataset details along with properties

Language	Gender	No.of Sentences	Words		syllables		phonemes	
			Total	Unique	Total	Unique	Total	Unique
Tamil	Male	1956	13,545	6159	48,049	1642	107,570	37
	Female	2335	15,880	6620	56,607	1696	126,659	37

calculate the 20 static, 20 delta and 20 acceleration coefficient using mel-frequency ceptral coefficients, appending an energy variable to each frame. The range of frequency is set to 0-8000 Hz with 40 mel bands. Delta and acceleration are calculated using width of 9 frames. In other words, each audio file is split it into frames using the Hamming windows function, and each frame is extracted to a feature vector of length 39.

4.3 Parameter settings

Our Proposed model is a 4-layer Deep Bidirectional Single Gated Unit (DBSGU) network [128 256 512 256 128] contains 320 cells respectively. Each layer is arranged in a sequence projection. Each hidden layer of bidirectional SGU incorporates one forward layer of SGU and one in reverse layer of SGU. Before training, samples are standardized to zero mean and unit deviation for each measurement. In the model, weights are introduced with a homogenous appropriation, and prepared utilizing pattern by analysing it statistically. We utilized a learning rate of 0.0005 and slope fragment condition per test sample of 0.0003. Early halting on the approval set is utilized to choose the best model. The foremost possible sequence of characters is produced by the model in a greedy manner. The last output grouping is then acquired by removing any blank symbols or reiterations of characters from the output and substitution any lower case letter with a space and its lowercase counterpart. output layer is divided into 2 softmax layer and hidden layer activation function are rectifier non-linearity. ADAM is used for cross entropy error in optimization [13], that run for 24 epochs and for regularization dropout is used. Instead of regular standard dropout, recurrent dropout is used to learn the long term dependencies. Input sentences for training the model is arranged in sorted order, and then based on the length of sentence it starts training the model from least length. Zero padding is minimized by using sentence sorting approach.

4.4 Tools and performance measures

The Kaldi toolbox is used for speech recognition [19]. The LibriSpeech formula was utilized for all examinations, including the evacuation, guidance and decoding of sound features [17]. The SRILM toolbox has been utilized for Language modelling [22]. ASR efficiency is anticipated by utilizing the Word Error Rate (WER) as the measurement.

5 Experimental results

The experiment was led on the CSMS data set. Our principle objective is to evaluate the quality of hybrid DBSGU-RNN to recognize large vocabulary continuous speech recognition, and specifically compare and contrast the methodology with already available DNN system and DBLSTM. The experiments are conveyed for DBLSTM, DBGRU and DBSGU based frameworks.

Table 3 Details of the training and testing data

Languages	Training		Testing		Final Testing
	Utterance (Male)	Utterance (Female)	Utterance (Male)	Utterance (Female)	Utterance (Male & Female)
Tamil	37,392	32,854	3,128	2,849	3,167

Table 4 Hybrid training results in %PER, %FER and %CE on the Tamil language

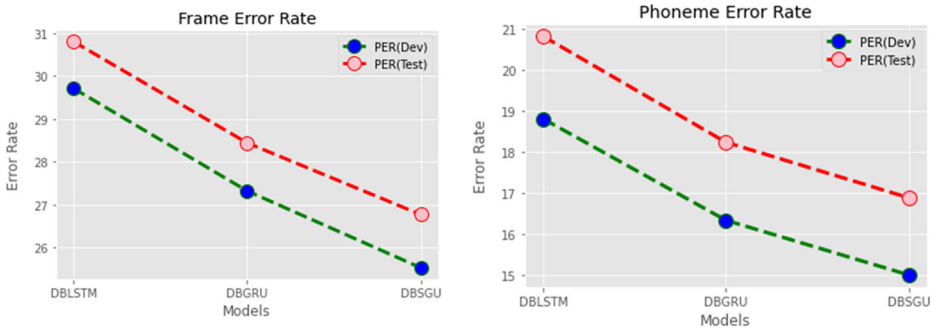
Network	Phoneme Error Rate (PER)		Frame Error Rate (FER)		Cross Entropy Error Rate (CE)	
	DEV	TEST	DEV	TEST	DEV	TEST
DBLSTM	18.80±0.21	20.81±0.34	29.71±0.30	30.80±0.46	0.96±0.09	1.01±0.013
DBGRU	16.33±0.143	18.23±0.14	27.32±0.13	28.44±0.30	0.82±0.010	0.87±0.018
DBSGU	15.00±0.14	16.88±0.12	25.53±0.07	26.77±0.15	0.77±0.007	0.82±0.003

The accuracy is computed using Eq. (20). A high-accuracy value represents maximized speech recognition performance.

$$Accuracy(\%) = \frac{No.\ of\ words\ are\ correctly\ recognized}{Total\ No.\ of\ words} \times 100 \tag{20}$$

Phoneme Error Rate (PER), Frame Error Rate (FER) and Cross Entropy error rate (CE) are demonstrated in Table 2 for DBSGU and DBLSTM framework. In DBLSTM we fixed 4 bidirectional networks along with 400 tanh gate in each, giving it about indistinguishable measure of weights as the DBSGU systems.

From the Table 3, it is clear that the proposed DBSGU performed well compared with DBLSTM. The SGU is the type of GRU without reset gate. The performance of proposed DBSGU is comparatively same as DBGRU. Recognizing the long dependency speech signal is addressed in the most crucial way by removing the reset gate. The learning rate of the model is also increased around 30%, to train the model with DBGRU it takes nearly 42 min but the removed reset gate model to learn the features within 24 min per epoch (Table 4).



(a) Phoneme Error Rate (Tamil) **(b) Frame Error Rate (Tamil)**

Fig. 8 (**a**) Phoneme error rate (Tamil). (**b**) Frame error rate (Tamil)

Table 5 Epoch with average accuracy

Model	Hidden layers	Cells (or) Neuron's	Learning rate	Epoch	Average accuracy
DBSGU	2	128	0.0005	980	48.20%
	2	256	0.0005	770	69.00%
	2	512	0.0005	1000	52.80%
DBSGU	3	128	0.0005	910	58.50%
	3	256	0.0005	680	75.30%
	3	512	0.0005	540	65.40%
DBSGU	4	128	0.0005	950	84.75%
	4	256	0.0005	810	88.40%
	4	512	0.0005	1000	82.49%

As shown in the Fig. 7, our proposed methodology outperforms the DBLSTM technique (baseline procedure) with and without dynamic features (Fig. 8).

Stochastic gradient descent were used to train a DBLSTM, initially we fixed 0.1 as a learning rate and 0.9 as a momentum. The proposed DBSGU system performance well when compared to DBGRU and DBLSTM. We used the LibriSpeech formula in our methodology. Our findings are shown in Table 1. It is noted that Word Error Rate (WER) is considerably decreased when compared to DBLSTM and DBGRU (Table 5).

From the above Table 3, it is clear that the greatest precision achieved is 88.40% during the DBSGU model has 4 layers, 256 cells with learning rate of 0.0005 with 810 epochs. From the Table 3, we concluded that when we increase the number of layers, the average accuracy also increase consistently. We also tried to increase and decrease the learning rate but the average accuracy is decreased in both scenarios (Fig. 9).

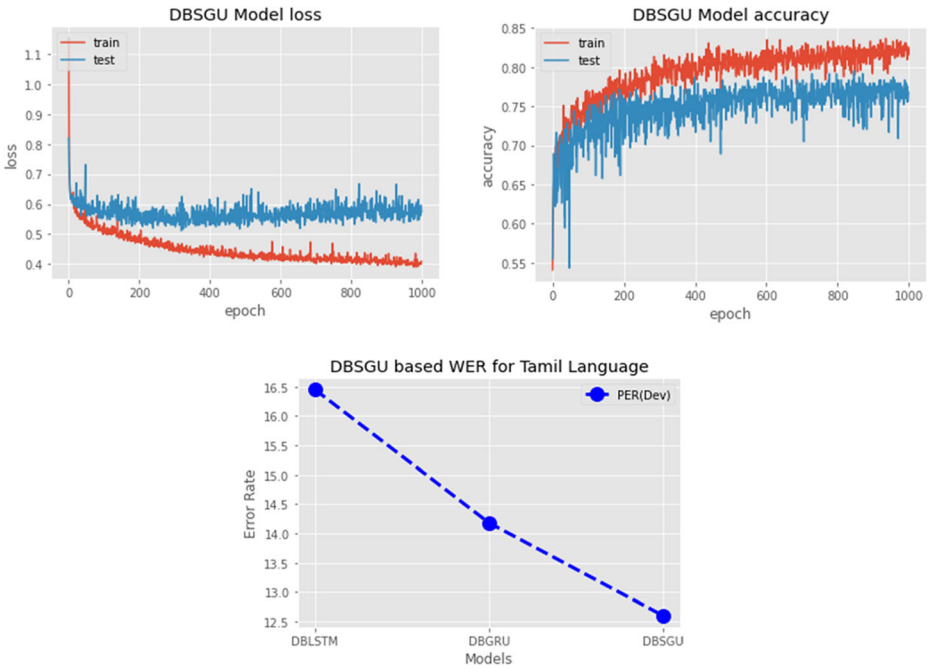


Fig. 9 DBSGU based WER for Tamil languages

Table 6 WER in % for Tamil languages

Language	DBLSTM	DBGRU	DBSGU
Tamil	16.45	14.18	**12.6**

Proposed model word error rate is highlighted in bold.

In Table 6, WER of the proposed model is compared with the DBLSTM and DBGRU. Accuracy is measured based on the total number of words present in the speech signal, the number of words are identified correctly. The proposed model runs more number of epochs when compared to DBGRU with the same amount of time taken for the training. For the experimental purpose various learning rate is used, such as 10^{-3}, 10^{-4} and 10^{-5}. Out of various learning rate 10^{-3} performed well with increased accuracy.

5.1 Performance

DBSGU takes less number of parameters for training compared with DBGRU and DBLSTM. Due to that DBSGU requires less amount of memory and training speed is also being increased. Figure 10 shows that the comparison between proposed DBSGU with DBLSTM and DBGRU. Based on the different batch size and input size, time taken for training the model is identified. The performance of proposed model is increased by fixing the batch size as 128 and input size as 256.

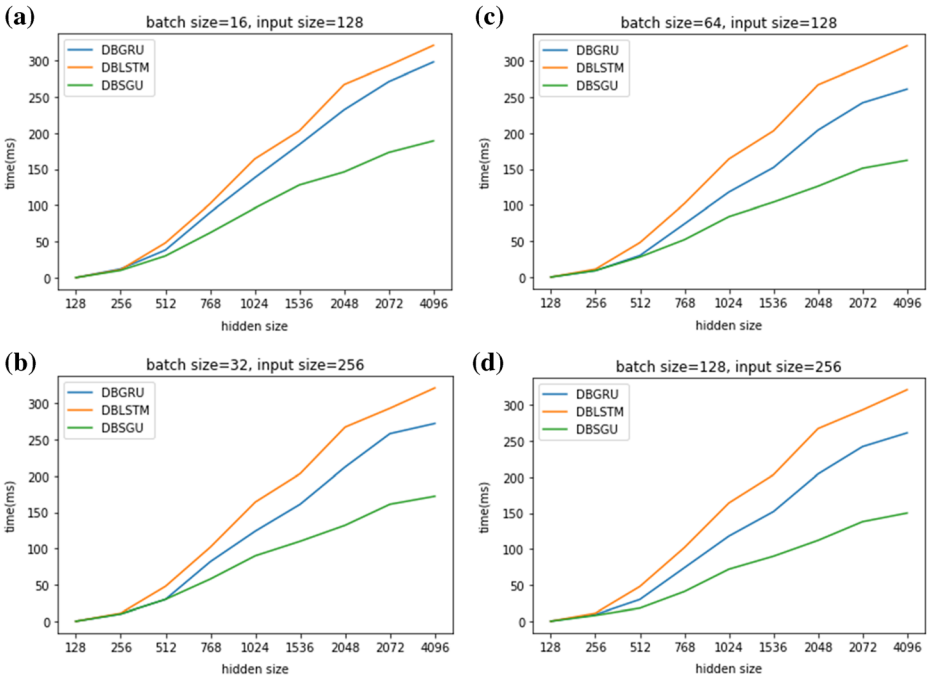


Fig. 10 (**a**), 10 (**b**), 10 (**c**) and 10 (**d**) shows the time taken to train the DBSGU model based on hidden size.

6 Conclusions

In this paper we have implemented the DBSGU, which is a Hybrid of DBNN with SGU in CSMS dataset for the word prediction from the given audio speech signal. We have compared the accuracy of our DBSGU model with the DBGRU and DBLSTM. From the results, it is seen that DBSGU could reach remarkably faster speed than the standard DBLSTM and achieves better performance. We also found that Word Error Rate (WER) is also decreased considerably for Tamil language. The proposed model is similar to DBGRU with removed reset gate, which increases the learning rate of model during training phase across 30% compared with DBLSTM. The performance of proposed system is similar to DBGRU. The proposed model maintains the accuracy even after the removal of reset with least amount of parameters. In future, the parameters can be increased to reduce training time f the model with better accuracy.

References

1. Abandah GA, Graves A, Al-Shagoor B, Arabiyat A, Al-Taee M (2015) Automatic diacritization of Arabic text using recurrent neural networks. Int J Doc Anal Recognit (IJDAR) 18(2):183–197
2. Barman PP, Boruah A (2018) A RNN based approach for next word prediction in Assamese phonetic transcription. Procedia Comput Sci 143:117–123 (ISSN 1877 – 0509)
3. Chavandan RS, Sable GS (2013) An overview of speech recognition using HMM. Int J Comput Sci Mob Comput 2(6):233–238
4. Chung J, Gulcehre C, Cho K, Bengio Y (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv arXiv:1412.3555
5. Cheng G, Povey D, Huang L, Xu J, Khudanpur S, Yan Y (2018) Output-gate projected gated recurrent unit for speech recognition. Interspeech, pp 1793–1797
6. Graves A, Schmidhuber J (2005) Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Netw 18(5–6):602–610
7. Graves A, Mohamed A, Hinton G (2013) Speech recognitionwith deep recurrent neural networks. In: Proc ICASSP 2013, Vancouver, Canada
8. Greff K, Srivastava RK, Koutnk J, Steunebrink BR, Schmidhuber J (2015) LSTM: A search space odyssey. arXiv: 1503.04069
9. He F, Chu SH, Kjartansson O, Rivera C, Katanova A, Gutkin A, Demirsahin I, Johny C, Jansche M, Sain S et al (2020) Open-source Multi-speaker Speech Corpora for Building Gujarati, Kannada, Malayalam, Marathi, Tamil and Telugu Speech Synthesis Systems. In: Proceedings of the 12th LREC Conference, Marseille, France, 11–16
10. Hochreiter S, Jürgen S (1997) Long short-term memory. Neural Comput 9:1735–80. https://doi.org/10.1162/neco.1997.9.8.1735
11. Jozefowicz R, Zaremba W, Sutskever I (2015) An empirical exploration of recurrent network architectures. In Proceedings of the 32nd International Conference on Machine Learning, Lille, France, vol 37, pp 2342–2350
12. Kim J, Kim J, Thu HLT, Kim H (2016) Long short term memory recurrent neural network classifier for intrusion detection. International Conference on Platform Technology and Service (PlatCon), Jeju, pp 1–5
13. Kingma DP, Ba J (2014) Adam: A method for stochastic optimization, CoRR, vol abs/1412.6980
14. Kumar J, Goomer R, Singh AK (2018) Long Short Term Memory Recurrent Neural Network (LSTM-RNN) based workload forecasting model for cloud datacenters. Procedia Comput Sci 125:676–682 (**ISSN 1877 – 0509**)
15. Kumar S, Hussain L, Banarjee S, Reza M (2018) Energy load forecasting using deep learning approach-LSTM and GRU in spark cluster. Fifth International Conference on Emerging Applications of Information Technology (EAIT), Kolkata, pp 1–4
16. Li X, Xianyu H, Tian J, Chen W, Meng F, Xu M et al (2016) A deep bidirectional long short-term memory based multi-scale approach for music dynamic emotion prediction. In: IEEE International Conference in Acoustics, Speech and Signal Processing (ICASSP); Shanghai, China, p 544–548

17. Panayotov V, Chen G, Povey D, Khudanpur S. Librispeech: an ASR corpus based on public domain audio books. In: International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Brisbane, Australia, pp 5206–5210
18. Panzner M, Cimiano P (2016) Comparing hidden Markov models and long short term memory neural networks for learning action representations. In: Pardalos P, Conca P, Giuffrida G, Nicosia G (eds) Machine Learning, Optimization, and Big Data. MOD 2016, vol 10122. Springer, Cham
19. Povey D, Ghoshal A, Boulianne G, Goel N, Hannemann M, Qian Y, Schwarz P, Stemmer G (2011) The kaldi speech recognitiontoolkit. In: Workshop on Automatic Speech Recognition and Understanding (ASRU), Hawaii, US, pp 1–4
20. Ravanelli M, Brakel P, Omologo M, Bengio Y (2018) Light gated recurrent units for speech recognition. IEEE Trans Emerg Top Comput Intell 2(2):92–102
21. Schuster M, Paliwal KK (1997) Bidirectional recurrent neural networks. Signal Process IEEE Trans 45(11): 2673–2681
22. Stolcke A (2002) SRILM-An extensible language modeling toolkit. In: International Conference on Spoken Language Processing (ICSLP), Denver, Colorado, pp 901–904
23. Thireou T, Reczko M (2007) Bidirectional Long Short-Term Memory Networks for Predicting the Subcellular Localization of Eukaryotic Proteins. IEEE/ACM Trans Comput Biol Bioinform 4(3):441–446
24. Zhang Y, Chen G, Yu D, Yao K, Khudanpur S, Glass JR (2016) Highway long short-term memory RNNS for distant speech recognition. In: Proc. of ICASSP 2016, pp 5755–5759
25. Zhou G-B, Wu J, Zhang C-L, Zhou Z-H (2016) Minimal gated unit for recurrent neural networks. Int J Automat Comput 13(3):226–234