STABILISING AND ACCELERATING LIGHT GATED RECURRENT UNITS FOR AUTOMATIC SPEECH RECOGNITION

Adel Moumen, Titouan Parcollet

STABILISING AND ACCELERATING LIGHT GATED RECURRENT UNITS FOR AUTOMATIC SPEECH RECOGNITION

Adel Moumen[1]*, Titouan Parcollet*[1,2]

[1]Laboratoire Informatique d'Avignon, Avignon Université
[2]Cambridge Machine Learning Systems Lab, University of Cambridge

ABSTRACT

The light gated recurrent units (Li-GRU) is well-known for achieving impressive results in automatic speech recognition (ASR) tasks while being lighter and faster to train than a standard gated recurrent units (GRU). However, the unbounded nature of its rectified linear unit on the candidate recurrent gate induces an important gradient exploding phenomenon disrupting the training process and preventing it from being applied to famous datasets. In this paper, we theoretically and empirically derive the necessary conditions for its stability as well as engineering mechanisms to speed up by a factor of five its training time, hence introducing a novel version of this architecture named SLi-GRU. Then, we evaluate its performance both on a toy task illustrating its newly acquired capabilities and a set of three different ASR datasets demonstrating lower word error rates compared to more complex recurrent neural networks.

Index Terms— Speech Recognition, Recurrent Units.

1. INTRODUCTION

Automatic Speech Recognition (ASR) is the task of transforming spoken language into text. Myriads of real-life devices and applications ranging from personal assistants to smart cars and automatic captioning rely on advanced ASR systems to provide the optimal experience to end users. ASR has greatly benefited from the rise of deep learning to improved robustness and performance. In practice, various deep learning techniques and models exhibiting different strengths and weaknesses have been developed over the last decade. Transformer neural networks, for instance, have reached unprecedented levels of word error rates for offline ASR [1]. Sequence-to-sequence (seq2seq) modelling with recurrent neural networks (RNN), on the other hand, has steadily achieved top-notch performance both for online and offline ASR while remaining fairly simpler to approach and implement both for academia and for the industry. Nowadays, transformers and RNN still co-exist widely, and the final choice of the architecture boils down to the desired use-case and the available compute resources. A successful implementation of a recurrent CTC-Attention encoder-decoder was first proposed in [2], and consisted in a recurrent encoder with an attentional recurrent decoder trained jointly with the CTC loss function [3, 4]. Such encoder-decoder architectures offer a rich literature as well as an ever-growing number of open-source implementations [5, 6] which continue to power many real-life products relying on ASR.

Following the CTC-Attention paradigm, one must find the most appropriate encoder to turn the speech signal into a latent subspace that the decoder can further process. In practice, and based on the successful DeepSpeech architecture [7], many competitive encoders simply combine convolutional layers with a deep recurrent neural network. Hence, the choice of the recurrent unit is of crucial interest to achieve state-of-the-art word error rates. For instance, the light gated recurrent units (Li-GRU) [8] network has been designed to carefully address the task of ASR. A Li-GRU is a compact single-gate unit derived from the gated recurrent units (GRU) which reduce by 30% the per-epoch training time over a standard GRU while also improving the ASR accuracy. Nevertheless, and despite a clear interest from the community, two major issues prevent a stronger adoption of the Li-GRU: (1) it highly suffers from exploding gradients as the gate is unbounded; and (2) no optimized implementation exists, hence leading to much larger training times than more complex alternatives such as LSTM neural networks.

The problems of vanishing and exploding gradients in recurrent neural networks (RNNs) have been studied for decades and several methods have been proposed. Gradient clipping and weight decay, for instance, directly act on the values that are flowing throughout the network during the forward and backward propagations [9, 10]. Regularization techniques, such as the soft orthogonal regularisation (SOR) [11] constraint, propose to directly tackle the problem of uncontrolled growth of the error accumulation during backpropagation through time (BPTT) [12] by introducing a term in the loss enforcing the recurrent weight matrices to remain orthogonal [11]. Others even proposed to develop a novel RNN architecture learning a unitary recurrent weight matrix in the complex domain [13]. An even more stable approach is to directly clip the eigenvalues of the recurrent weight matrices as soon as they reach a certain threshold [14]. However, the two latest solutions introduce important complexity overheads at computation time and remain mostly intractable for ASR scenarios with medium and large datasets. Then, and despite promising research directions, the instability of the Li-GRU has never been formally or empirically investigated. It remains impossible to verify if existing methods preventing exploding gradients could benefit the Li-GRU.

This article proposes to formalise and solve the issue that the original Li-GRU encounters. Hence, it introduces the Stabilised Li-GRU (SLi-GRU), offering theoretical and empirical guarantees on its stability compared to the original Li-GRU as well as a five times reduced training time following a well-designed CUDA implementation. In summary, the contributions are fourfold: 1. Theoretically ground the conditions to avoid exploding gradients with the Li-GRU; 2. Analyse the impact of existing methods both theoretically and empirically with the additive task on the Li-GRU, leading to the introduction of the SLi-GRU; 3. Empirically validate the scaling of the SLi-GRU to three well-known speech recognition tasks; and 4. Deliver to the community a CUDA optimized version of the Li-GRU and SLi-GRU within SpeechBrain[1].

The conducted experiments on LibriSpeech and CommonVoice

[1]https://speechbrain.github.io/

demonstrate that the SLi-GRU is much more stable and faster than the original Li-GRU which can not be trained with such datasets. It also obtains better ASR performance than the state-of-the-art baseline offered within Speechbrain based on LSTM neural networks.

2. ON THE INSTABILITY OF THE LI-GRU

The initial definition of the Li-GRU [8] is unstable and, in practice, can not be applied to even medium-sized speech recognition datasets such as Librispeech or CommonVoice. First, we recall the basic equations of the Li-GRU (Section 2.1). Then, following a theoretical analysis of the gradient exploding phenomenon arising at training time, we derive the necessary bounds to ensure a smooth training (Section 2.2). Finally, we formalise various approaches to tackle the gradient exploding problem by investigating theoretically their impact on the later boundaries (Section 2.3).

2.1. Light gated recurrent units

The Li-GRU [8] has been designed to carefully address the task of speech recognition efficiently. By removing the reset gate, adding a ReLU on the candidate gate, and applying batch normalization ($\mathcal{BN}$) [15] on the feed-forward connections the authors were not only able to reduce by 30% the per-epoch training time of a Li-GRU against a standard GRU [16], but also to improve the performance across different tasks. In particular, the Li-GRU equations are:

$$z_t = \sigma(\mathcal{BN}(W_z x_t) + U_z h_{t-1}), \tag{1}$$

$$\tilde{h} = \text{ReLU}(\mathcal{BN}(W_h x_t) + U_h h_{t-1}), \tag{2}$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}. \tag{3}$$

with z_t the update gate, $\tilde{h_t}$ the candidate gate, h_t the hidden state, all taken at the time step t, and h_{t-1} the hidden state from the previous time step. The logistic sigmoid function is denoted as σ while the operator $\odot$ refers to the element-wise product. The Li-GRU is fed at each time step with a vector x_t. W_z, W_h are the feed-forward, and U_z, U_h the recurrent weights. In the following, we consider W_* and U_* as references to the weights independently of the gates.

2.2. Gradient instabilities and temporal contributions

Despite its simplicity, the Li-GRU exhibits a weakness in Eq. 2 as the recurrent process is unbounded, and therefore subject to potential gradient instabilities. Here, and following carefully previous work from [10, 11, 13, 17], we propose to study from an analytical point of view the theoretical motivations of the Li-GRU instability.

Let $E = \sum_N^{t=0} E_t$ be the total loss, where N is the sequence length, and E_t the loss at the time step t. We can highlight the gradient exploding problem in the Li-GRU with the BPTT by showing that each backpropagation step in the recurrent process is controlled by a value η that may explode as soon as it deviates from 1. Following the BPTT, we first obtain $\partial E_t / \partial h_t$ and then backpropagate from $\partial E_t / \partial h_m$ to $\partial E_t / \partial h_{m-1}$ with ($m \leq t$):

$$\frac{\partial E_t}{\partial h_m} = \frac{\partial E_t}{\partial h_t} (\prod_{i=m}^{t} \frac{\partial h_i}{\partial h_{i-1}}), \tag{4}$$

$$\frac{\partial E_t}{\partial h_m} = \frac{\partial E_t}{\partial h_t} (\prod_{i=m}^{t} \frac{\partial^+ h_i}{\partial h_{i-1}} + \frac{\partial h_i}{\partial z_i}\frac{\partial z_i}{\partial h_{i-1}} + \frac{\partial h_i}{\partial \tilde{h_i}}\frac{\partial \tilde{h_i}}{\partial h_{i-1}}), \tag{5}$$

with $\partial^+ h_i / \partial h_{i-1}$ refering to the immediate partial derivative. Then, we derive the $\partial h_i / \partial h_{i-1}$ upper bound called η that will describe the necessary conditions triggering the explosion phenomenon. η is obtained by computing the norm of each partial derivatives of $\partial h_i / \partial h_{i-1}$:

$$\eta = \frac{\gamma_1}{4} ||U_z||_2 + ||U_h||_2, \tag{6}$$

with $\gamma_1 = max_{1 \leq m \leq t, 1 \leq j \leq d} |[h_{m-1}]_j|$, and d the hidden dimension size. Then, considering $||\partial E_t / \partial h_{m-1}||$ and $||\partial E_t / \partial h_m||$ at adjacent timesteps and leveraging the norm properties:

$$||\frac{\partial E_t}{\partial h_{m-1}}|| \leq \eta ||\frac{\partial E_t}{\partial h_m}||. \tag{7}$$

Finally, by induction, we generalize over non adjacent time steps noted p subject to $p < m$ (*i.e* applying η, $m - p$ times):

$$||\frac{\partial E_t}{\partial h_p}|| \leq \eta^{m-p} ||\frac{\partial E_t}{\partial h_m}||. \tag{8}$$

It results that the Li-GRU may explode as soon as $\eta > 1$ following the increase of the $m - p$ quantity, hence, leading to the exploding gradients phenomenon. Therefore, we wish to keep η as close as possible to 1.

2.3. Stabilizing the Li-GRU

This section presents various methods to tackle exploding gradients in the Li-GRU by analysing their impacts on η.

Soft orthogonal regularization (SOR). The SOR [11] helps any RNN to keep U_* close to orthogonal by adding a regularization term to the loss, hence limiting the impact of U_* on η. The spectral norm of a matrix $||W||_2$ is 1, simplifying the value of η from Eq. 6:

$$\eta = \frac{\gamma_1}{4} + 1. \tag{9}$$

Despite being helpful to approach the problem of exploding gradients, such a regularization will constrain the weights to be in the Stiefel manifolds, and potentially degrade the global training loss.

Weight decay and gradient clipping. Weight decay (WD) [9] adds a penalty on the l2-norm of U_* and W_* potentially reducing the impact of the recurrent weights on η. Nevertheless, this method is not always sufficient and often is coupled with gradient clipping (GC) [10, 18] that rescales the gradient so that its norm remains lower than a threshold, hence postponing the growth of U_*. In practice, this method is effective for U_h that is never rescaled.

Sine activation function. In [19], the authors demonstrated that the sine activation function was an excellent bounded alternative for recurrent neural networks. In our case, it will be the non-linearity of the candidate gate and therefore modify the expression of η as:

$$\eta = \frac{\gamma_1}{4} ||U_z||_2 + \cos||U_h||_2. \tag{10}$$

Such a change is effective in reducing the impact of $||U_h||_2$ on η, however, it remains that γ_1 may explode.

Layer normalization and recurrent weights. Layer normalization ($\mathcal{LN}$) [20] normalizes its input to have zero mean and unit variance. If applied to the product of U_* and h_{t-1} in Eq. 2 and Eq. 1, it will also reduce the impact of γ_1 on η, rescale the gradient of U_*, and accelerate the convergence of the Li-GRU. Introducing the $\mathcal{LN}$ to the original Li-GRU leads to a novel Li-GRU:

$$z_t = \sigma(\mathcal{BN}(W_z x_t) + \mathcal{LN}(U_z h_{t-1})), \tag{11}$$

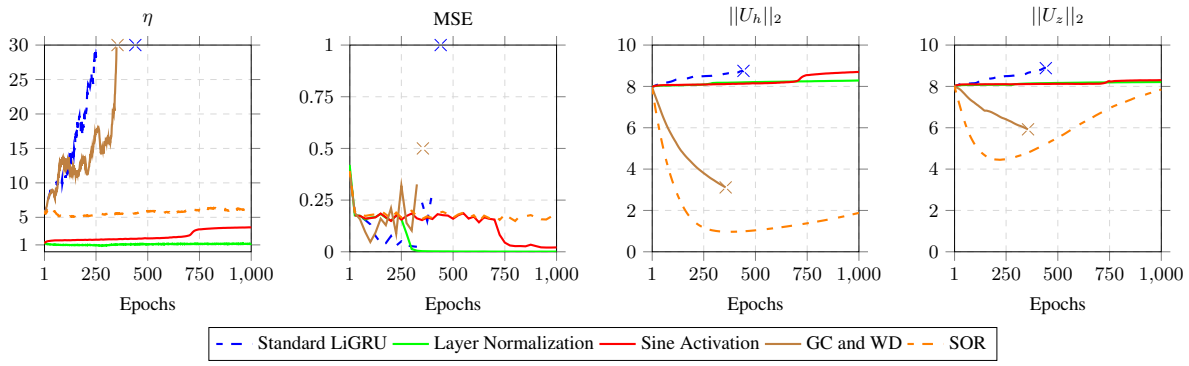

Fig. 1. Results of the adding task with a sequence length of $2,000$ with different Li-GRUs equipped with different methods to prevent gradient exploding to happen (Section 2.3). η is the scale of the gradient that transports the error in time. It must remain close to 1. The MSE loss function must be close to 0. $||U_z||_2$ and $||U_h||_2$ shall remain as stable or small as possible. "GC and WD" refers to a Li-GRU with weight decay and gradient clipping while "SOR" is the soft orthogonal constraint. A small coloured cross indicates that gradients exploded. The layer normalization is the only method that solves the issue by keeping η close to 1, and at the same time, exhibiting the best MSE.

$$\tilde{h} = \text{ReLU}(\mathcal{BN}(W_h x_t) + \mathcal{LN}(U_h h_{t-1})), \tag{12}$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}. \tag{13}$$

Following the proof from [17], η becomes:

$$\eta = \frac{\gamma_1}{4\sigma_z}||U_z||_2 + \frac{1}{\sigma_h}||U_h||_2. \tag{14}$$

with σ the standard deviation of the corresponding gates. In the latter formulation, and motivated by [21], we decided to get rid of the gain and bias terms of the layer normaliation to even further simplify the architecture. In short, a simple $\mathcal{LN}$ enables $||\partial E_t/\partial h_m||$ to remain unaffected by the scaling of the recurrent weights but also minimizes the value of γ_1 due to the normalization of the input.

3. EMPIRICAL EVIDENCES

The empirical validity of the different proposed solutions to the exploding gradients phenomenon is first investigated to properly identify a successor of the Li-GRU named Stabilized Li-GRU (SLi-GRU). Hence, we monitor various metrics including η during a task designed to rapidly trigger large gradients (Section 3.1). Then, the SLi-GRU is compared to a standard Li-GRU and well-known LSTM networks on three datasets of speech recognition (Section 3.2).

3.1. The adding task

The adding task [22] is a synthetic benchmark designed to generate long sequences quickly highlighting exploding gradients. In practice, a neural network must learn to predict the addition of two numbers that are stored in large sequences. Indeed, having long sequences increases the probability of seeing $\partial h_t/\partial h_m$ explode. Then, the task requires the network to store real values for very long periods, hence giving a convenient way of evaluating the network memory.

Task Definition. Each input sequence is designed as a pair of components. The first component is a real value sampled from a uniform distribution in the range $[0, 1]$ while the second is either 1 or 0, and is used as a marker. The goal of the network in this task is to remember the values of the first component of the two elements that have a 1 in the second component so that the sum can be produced at the output. All input sequences have a length of T. The first marker equal to 1.0 is randomly picked in the range $[0, T/2 - 1]$ while the other is in the range $[T/2, T - 1]$. Using a large value of T (*e.g* $2,000$) is sufficient to create the exploding gradient problem and to challenge the network to do a correct summation.

Architecture Details and Network Training. All models are trained following the same setup so that the effect of the different techniques may be properly compared. The maximum number of epochs is $1,000$, and the number of time steps T is $2,000$. The batch size is 256. Adam [23] is used as the optimizer with a learning rate of 0.001. Each Li-GRU configuration used a ReLU if not mentioned in the legend and a single hidden layer with $1,024$ hidden units. The soft orthogonal regularization λ is set to 0.001 after carefully trying multiple values. Weight decay is fixed to 0.001 as well and the gradient clipping norm threshold is given at 1. The training loss is the standard mean squared error (MSE) and the recurrent weights follow an orthogonal initialization [24].

Results. As shown in Fig. 1, the layer normalization outperformed all the other methods on the principal metrics (i.e η and MSE). Not only the $\mathcal{LN}$ keep η close to 1, but also converged faster to an excellent MSE of 0.0005 whereas the sine activation, *i.e* the second best, achieved an MSE of 0.01. The sine-based Li-GRU is also slower than its $\mathcal{LN}$ counterpart to converge. Moreover, the value of η deviated from 1 with the sine Li-GRU as the number of epochs reached 700 indicating the start of an instability in the gradients. The SOR, on the other hand, and despite promising performance to postpone the problem of exploding gradients, was not able to reach a correct MSE due to a negative effect of the constraint applied to the recurrent weights. The standard Li-GRU and its gradient clipping and weight decay variant were not able to prevent the exploding gradient phenomenon and crashed at the same time than the standard Li-GRU. The experiments costed a total of 0.04 kg of CO_2 [25].

Introducing the SLi-GRU. This analysis demonstrates that the layer normalized Li-GRU, named SLi-GRU thereafter, is the only architecture to: (1) remember past information, hence leading to an excellent MSE score, (2) postpone the gradient exploding problem successfully under tight conditions, and (3) offer a theoretical formulation of η that satisfies the need for a high level of stability.

Table 1. Results of different CRDNN equipped with different RNN expressed in terms of word error rate (WER) and character error rate (CER) (*i.e.* lower is better) on the test sets of CommonVoice French, Italian and LibriSpeech. *Inf* refers to infinity *i.e.* exploding gradients. **Params (M)** is the number of neural parameters. Results in bold are the best results over the experiments. As expected the original Li-GRU exploded on all tasks while the SLi-GRU obtains the best performance even compared to the best LSTM-based CRDNN from SpeechBrain.

	CV Italian			**CV French**			**LS 960 Clean**			**LS 960 Other**		
Method	**CER %**	**WER %**	**Params (M)**	**CER %**	**WER %**	**Params (M)**	**CER %**	**WER %**	**Params (M)**	**CER %**	**WER %**	**Params (M)**
LSTM	3.50	11.06	148.2	5.86	14.61	148.2	1.87	3.83	173.0	6.0	10.36	173.0
SLi-GRU	**3.05**	**10.05**	148.1	**5.32**	**13.46**	148.1	**1.27**	**3.42**	170.0	**3.77**	**7.82**	170.0
LSTM	6.82	16.28	50.8	7.46	18.96	50.8	2.83	4.74	94.8	8.48	12.84	94.8
SLi-GRU	3.56	11.56	50.6	6.34	15.53	50.6	2.04	4.06	94.3	5.96	10.41	94.3
LiGRU	*Inf*	*Inf*	148.1	*Inf*	*Inf*	148.1	*Inf*	*Inf*	170.0	*Inf*	*Inf*	170.0
LiGRU	*Inf*	*Inf*	50.6	*Inf*	*Inf*	50.6	*Inf*	*Inf*	94.3	*Inf*	*Inf*	94.3

3.2. Automatic speech recognition experiments

The task of speech recognition is to transcribe the content of an audio signal into a human-readable output. One of the challenges of speech recognition lies in the long input sequences potentially triggering exploding gradients with RNNs. In the following, and to illustrate the latter issue, we compare our newly introduced SLi-GRU to Li-GRU and LSTM neural networks on three different datasets including CommonVoice French, Italian and LibriSpeech.

Datasets. Three datasets with different complexities and sizes are considered to evaluate the models: LibriSpeech [26], CommonVoice French (version 8.0) [27], and CommonVoice Italian (version 8.0) [27]. These datasets are primer choices as they exhibit long sequences making the training challenging for the SLi-GRU while remaining extremely competitive as the community has been extensively investigating them. The utterances of CommonVoice (CV) are obtained from volunteers all around the world. The French set (CV-fr) contains 460K utterances (671 hours) with different accents, and more than 16K participants. The train set consists of 620.59 hours, while both validation and test sets contain 25.52, and 25.52 hours of speech respectively. The Italian set (CV-it), is relatively small compared to CV-fr. It contains 208, 24, 26 hours training, validation, and test data. Finally, LibriSpeech (LS) is a corpus of 960 hours of read and clean English speech.

Architecture Details and Network Training. We trained an LSTM [28], a Li-GRU, and a SLi-GRU with two different scenarios on a single and popular end-to-end (E2E) ASR architecture with all datasets: an encoder-decoder CTC-Attention based CRDNN coming from the SpeechBrain toolkit [5]. The encoder is composed of three distinct parts: a VGG-like features extractor, a bidirectional RNN, and a deep dense neural network. This is combined with a location-aware attentive GRU decoder jointly trained with the CTC loss. The two different scenarios that are considered for the Li-GRU are: (1) a *low-budget model (LB)*, as claimed by the original Li-GRU authors, we wished to verify if a smaller Li-GRU could achieve state-of-the-art performance. In this particular case, the ASR system has 50.6M and 94.3M parameters on CV and LS respectively; (2) a *high-budget model (HB)*, with the Li-GRU matching approximately the same number of neural parameters than the best LSTM-based CRDNN from SpeechBrain (*i.e* 148M and 170M parameters). Hyperparameters and neural architectures vary across the different datasets and are extensively described in the corresponding *SpeechBrain* recipes [5] (commit hash *ee50231*). A recurrent language model (LM) trained with the LS language modelling resources is coupled via shallow fusion for LibriSpeech. No LM is used with CV.

Results. First, it is worth underlining that the standard Li-GRU did not even succeed in lasting more than one epoch on all tasks as the gradient exploding problem formally demonstrated in this work simply hits hard empirically as well. Such a problem is solved with the introduction of our SLi-GRU that was able to proceed for the entire training hence proving that the recurrent layer normalization is an effective solution to approach exploding gradients in RNNs. Then, with a fixed size of 148M and 170M parameters, the SLi-GRU outperformed the previous state-of-the-art CRDNN model from SpeechBrain on the three datasets with WER of 7.82%, 3.42%, 13.46% and 10.05% compared to 10.36%, 3.83%, 14.61% and 11.06% for the LSTM on the LS other and clean, and CV French and Italian datasets respectively. Even more interestingly, the *LB* SLi-GRU solely equipped with a third and half of the neural parameters on CV and LS reached promising performance as the WER only degraded by 0.425 in average compared to the *HB* LSTM over the three datasets. Furthermore, the *LB* SLi-GRU reduced the WER by 21% in average against the *LB* LSTM. Following the original findings of the Li-GRU [8], it seems clear that our SLi-GRU represents an interesting alternative to LSTM for speech recognition in *HB* and *LB* scenarios. Finally, we estimate that at least 39.3 kg of CO_2 have been emitted to produce these results [25].

3.3. Efficiency perspectives

The original Li-GRU made available to the community did not leverage any optimization scheme at training time. Hence, it introduced an extremely large overhead in iteration time, even compared to much more complex LSTM networks that directly benefit from a CUDA implementation. As we wish our method to impact widely and positively the community, we decided to implement a custom CUDA version of the SLi-GRU so that anyone may use it with large datasets without suffering from excessive training times. The latter contribution is five times faster and change the time-complexity from a quadratic to a linear regime, compared to the default PyTorch implementation. This contribution replaces the previous Li-GRU implementation in the well-known SpeechBrain toolkit.

4. CONCLUSION

We introduced the SLi-GRU, a theoretically grounded and empirically validated stabilized version of the original Li-GRU. Following an analytical demonstration, we showed that applying a layer-wise normalization to the recurrent gate of the SLi-GRU is sufficient to prevent the exploding gradients phenomenon. During the conducted experiments, the SLi-GRU outperformed the previous state-of-the-art relying on LSTM from the SpeechBrain toolkit in three automatic speech recognition tasks. Furthermore, we also released an optimized version of the SLi-GRU further reducing the required training time by a factor of five.

5. REFERENCES

[1] Shigeki Karita, Nanxin Chen, Tomoki Hayashi, Takaaki Hori, Hirofumi Inaguma, Ziyan Jiang, Masao Someki, Nelson Enrique Yalta Soplin, Ryuichi Yamamoto, Xiaofei Wang, et al., "A comparative study on transformer vs rnn in speech applications," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 449–456.

[2] Shinji Watanabe, Takaaki Hori, Suyoun Kim, John R Hershey, and Tomoki Hayashi, "Hybrid ctc/attention architecture for end-to-end speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.

[3] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.

[4] Alex Graves and Navdeep Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *International conference on machine learning*. PMLR, 2014, pp. 1764–1772.

[5] Mirco Ravanelli, Titouan Parcollet, Peter Plantinga, Aku Rouhe, Samuele Cornell, Loren Lugosch, Cem Subakan, Nauman Dawalatabad, Abdelwahab Heba, Jianyuan Zhong, et al., "Speechbrain: A general-purpose speech toolkit," *arXiv preprint arXiv:2106.04624*, 2021.

[6] Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson-Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, et al., "Espnet: End-to-end speech processing toolkit," *Proc. Interspeech 2018*, pp. 2207–2211, 2018.

[7] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al., "Deep speech 2: End-to-end speech recognition in english and mandarin," in *International conference on machine learning*. PMLR, 2016, pp. 173–182.

[8] Mirco Ravanelli, Philemon Brakel, Maurizio Omologo, and Yoshua Bengio, "Light gated recurrent units for speech recognition," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 2, pp. 92–102, 2018.

[9] Ilya Loshchilov and Frank Hutter, "Decoupled weight decay regularization," in *ICLR*, 2019.

[10] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio, "On the difficulty of training recurrent neural networks," in *International conference on machine learning*. PMLR, 2013, pp. 1310–1318.

[11] Eugene Vorontsov, Chiheb Trabelsi, Samuel Kadoury, and Chris Pal, "On orthogonality and learning recurrent networks with long term dependencies," in *International Conference on Machine Learning*. PMLR, 2017, pp. 3570–3578.

[12] Paul J Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[13] Martin Arjovsky, Amar Shah, and Yoshua Bengio, "Unitary evolution recurrent neural networks," in *International conference on machine learning*. PMLR, 2016, pp. 1120–1128.

[14] Sekitoshi Kanai, Yasuhiro Fujiwara, and Sotetsu Iwamura, "Preventing gradient explosions in gated recurrent units," in *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. 2017, vol. 30, Curran Associates, Inc.

[15] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. PMLR, 2015, pp. 448–456.

[16] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio, "On the properties of neural machine translation: Encoder–decoder approaches," *Syntax, Semantics and Structure in Statistical Translation*, p. 103, 2014.

[17] Lu Hou, Jinhua Zhu, James Kwok, Fei Gao, Tao Qin, and Tie-Yan Liu, "Normalization helps training of quantized lstm," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. 2019, vol. 32, Curran Associates, Inc.

[18] Tomáš Mikolov et al., "Statistical language models based on neural networks," *Presentation at Google, Mountain View, 2nd April*, vol. 80, no. 26, 2012.

[19] Giambattista Parascandolo, Heikki Huttunen, and Tuomas Virtanen, "Taming the waves: sine as activation function in deep neural networks," 2017.

[20] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton, "Layer normalization," *stat*, vol. 1050, pp. 21, 2016.

[21] Jingjing Xu, Xu Sun, Zhiyuan Zhang, Guangxiang Zhao, and Junyang Lin, "Understanding and improving layer normalization," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[22] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[23] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *ICLR (Poster)*, 2015.

[24] Andrew M. Saxe, James L. McClelland, and Surya Ganguli, "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks," *CoRR*, vol. abs/1312.6120, 2014.

[25] Titouan Parcollet and Mirco Ravanelli, "The Energy and Carbon Footprint of Training End-to-End Speech Recognizers," in *Proc. Interspeech 2021*, 2021, pp. 4583–4587.

[26] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[27] Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber, "Common voice: A massively-multilingual speech corpus," in *Proceedings of the 12th Language Resources and Evaluation Conference*, 2020, pp. 4218–4222.

[28] Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber, "Lstm: A search space odyssey," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 10, pp. 2222–2232, 2016.