COMPUTOLOGY: Journal of Applied Computer Science and Intelligent Technologies
Volume 1, Issue 1, Jan-Jun 2021, pp. 13-21
DOI: 10.17492/computology.v1i1.2102
https://www.journalpressindia.com/cjacsit

QLSTM-based Joint-Training for Noise Robust Hindi Speech Recognition

*Ankit Kumar**

ABSTRACT

In recent years, the field of speech recognition has benefited more from deep learning. The substantial improvement was reported by current technology; how- ever, speech recognition did not work well in a noisy environment. Improving speech recognition in noisy conditions is a critical task. The goal of this work is to propose a high accuracy noise-robust Hindi speech recognition system. In this series, we apply Bi-directional Quaternion Long-Short-Term Memory (QLSTM) neural network to train the speech enhancement and speech recognition model jointly. The role of the i-vector and Recurrent Neural Network (RNN) language model is also investigated. Using a 2.5-hour Hindi speech dataset and the Kaldi and Pytorch-Kaldi toolkit, all of the experiments were carried out. The proposed model reports the 2% Word Error Rate (WER) reduction over the state-of-the-art (SOTA) techniques.

Keywords: Quaternion neural network; Joint-training; Hindi speech recognition; Noise-robust ASR.

1.0 Introduction

Automatic Speech Recognition (ASR) is the task of computing text sequences of corresponding speech utterances. In recent years, the ASR field has witnessed a remarkable evolution, mainly because of deep-learning technologies [6]. Speech technologies are now commercially available at your fingertips, like Google and Alexa [6, 26]. A wide variety of DNN-based acoustic models available for use, in which RNN is well suited to sequential data like speech. However, RNN based models like LSTM, GRU require a huge number of parameters which increase the complexity of the ASR system. By and large, Neural Network models are over-parameterized, requiring a gigantic measure of computing power, memory, and less reasonable for low resource languages [22]. The neural network model with fewer parameters has received more attention in recent years [18, 16, 14, and 13]. On hardware that only has a limited amount of memory and computation power, neural network models with fewer parameters are simple to implement. These models are simple to train, take less time to train, and have a higher throughput of inference [9]. Except as noted above, the low-resource languages with few hours speech data for the neural network model refuse to use the many parameters to prevent over fitting [20]. When only a small amount of speech data is available, a neural network model with fewer parameters becomes the better option.

***Department of Computer Science & Information Technology, KIET Group of Institution, Ghaziabad, India (E-mail: anketvit@gmail.com)*

Recently, quaternion RNN (QRNN) [14] and QLSTM have been proposed with fewer parameter size. QL- STM is based on a hyper-complex number instead of a real-valued RNN. The QLSTM model has the ability to learn inter and intra-dependencies [12]. Except above, QLSTM has four times fewer parameters compared to the same size real-valued LSTM [14].

Despite this progress, there is still a vast gap in performance when acoustic condition changes [7]. To handle such adversity, various solutions were proposed, including speech enhancement [1], speech separation [10], and noise-robust features [2, 3]. The most common limitation of various techniques was the week matching and communication when more than one module integrated [17].

The combined form of speech enhancement with speech recognition helps to achieve some gain in accuracy. Generally, the speech enhancement module trained independently, and their metrics are not correlated with the speech recognition module, which leads to poor improvement in accuracy. Opposite to this, in the joint-training framework, we jointly train both modules and update their parameters together as they are a single big network [17].

In this work, we jointly trained speech enhancement and speech recognition module with the help of a recently proposed QLSTM [13] neural network. The joint- training framework was well investigated by various work [25, 24, 4, 5] mainly for the English language. For Hindi speech recognition, this would be the first attempt as per the best of our knowledge. The investigation of the QL- STM acoustic model on Hindi ASR is also not found in earlier work. The main obstacle to training the ASR sys- tem for most Indian languages, including Hindi, is the abundance of training data. A well-known research re- source is a three-hour Hindi dataset with a restricted vocabulary. To the best of our knowledge, Hindi ASR has not been systematically compared. For this reason, we pick just those articles which were prepared over a similar Hindi dataset for correlation.

The remaining part of the paper is organized as follows: Section 2 explains the quaternion algebra. Section 3 discusses the QLSTM model. Section 4 describes the joint-training framework. Section 5 gives the experimental setup and corpus details. Section 6 covers the experimental part of the paper, and Section 7 is the final conclusion of the proposed system.

2.0 Quaternion Algebra

A quaternions Q are the extension of the complex numbers. It is defined as:

$$Q = a1 + bi + cj + dk \quad (1)$$

where a, b, c, and d could be any real numbers and quaternion units denoted by 1, *i*, *j*, and *k*. In equation 1, a is the real part and $bi + cj + dk$ with $i2 = j2 = k2 = -1$ is the imaginary part. In quaternion, the real part can be scaler while imaginary part or vector part is a three-dimensional vector. Therefore, a quaternion can be defined as:

$$Q = (a, \vec{v}) \quad (2)$$

The conjugate Q is defined as:

$$Q = ai - bi - cj - dk \quad (3)$$

and unit quaternion Q is summarized as:

$$Q = \frac{Q}{\sqrt{a+b+c+d}} \quad (4)$$

The quaternion matrix Q stored the information in $4 * 4$ matrix of real numbers.

$$Q = \begin{bmatrix} a & -b & -c & -d \\ b & a & -d & c \\ c & d & a & -b \\ d & c & b & a \end{bmatrix} \quad (5)$$

The Hamilton product between two quaternions Q(a, b, c, d) and Q(a, b, c, d) can be easily done by using Q.

$$Q \otimes Q = (ae - bf - cg - dh) + (be + af - dg + ch).i + (ce + df + ag - bh).j + (de - cf + bg + ah).k \quad (6)$$

3.0 Quaternion Long-Short term Memory (QLSTM)

The QRNN and QLSTM were proposed by Parcollet et al. in 2018 for speech recognition. The quaternion LSTM is the variant of QRNN based on hyper-complex numbers. It is an extension of real-valued [11] and complex-valued [8, 23] RNN. All the parameters of the QLSTM model are quaternions. Due to the recurrent architecture, QL- STM has the ability to code the sequential dependencies, and due to the hamilton product, it can code internal dependencies [12]. In QLSTM, input and output vector dimensions are split into four parts: the first part is equal to a, the second one is equal to *bi*, the third one equal to *cj*, and the fourth one is equal to dk. In QLSTM, opposite to RNN, dot-product operations are replaced by the hamilton product. The QLSTM is the three gated architecture. It contains the input gate i, the output gate o, and the forget gate f. In addition to these three gates, c and h denote the cell-state and hidden state. The QLSTM can be summarized as:

$f = \sigma(W \otimes x + R \otimes h + b)$ (7)

$i = \sigma(W \otimes x + R \otimes h + b)$ (8)

$c = f * c + i * \alpha.(Wx + Rh + b)$ (9)

$o = \sigma(W \otimes x + R \otimes h + b)$ (10)

$h = o * \alpha.(c)$ (11)

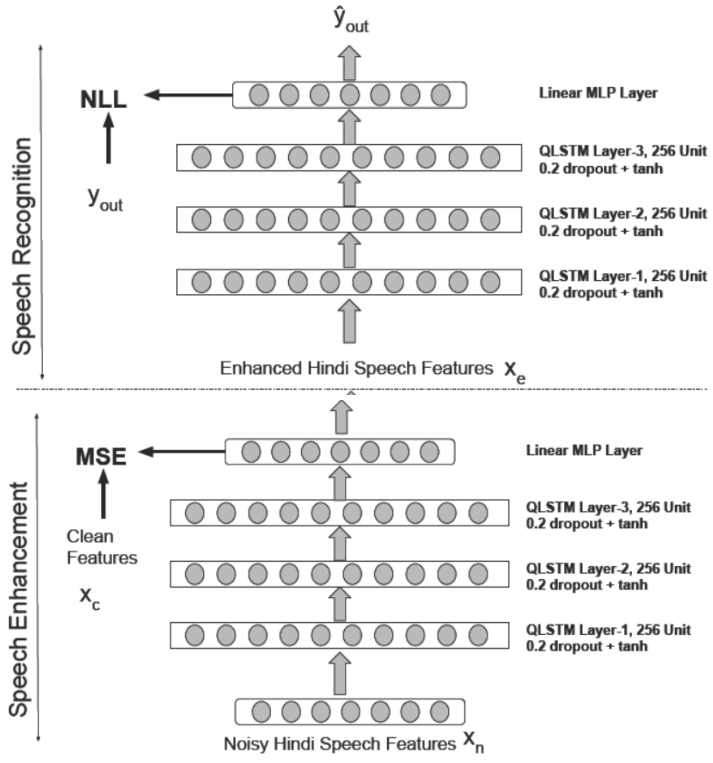
4.0 Joint Training Framework

A joint architecture was proposed by linking speech enhancement and a speech recognition QLSTM model. The upper part of the figure shows the speech recognition part, and the lower part highlights the speech enhancement. The noisy features feed as an input to the speech enhancement network. It will produce enhanced features set, which works as an input for the speech recognition neural network. In the joint training framework, first, perform the forward pass, estimate the loss function for each QLSTM model (Minimum square error, negative log- likelihood), compute the gradient, and back-propagate The Hamilton product between two quaternions Q(a, b, c, d) and Q(a, b, c, d) can be easily done by using Q. them.

The joint-training framework can be easily under- stood by the joint training algorithm described in Alg. 1, where x, x, x denote the noisy, enhances, and clean speech features, N is the number of samples in mini-batch, g denotes the gradient and θ denotes the QLSTM parameters. In

addition to this, η is the learning rate, and λ is the weighting hyperparameter for gsr. The parameters updation is taken place as:

Figure 1: The Proposed QLSTM Architecture for Joint-Training



Alg 1: Pseudo-code for Joint-training using Quaternion Neural Network

Quaternion LSTM weight initialization
for each mini-batch j do:
Forward-pass:
Do a forward pass through the network using $_{n}$ input.
Compute the Speech Enhancement (SE) & Speech Recognition (SR) cost functions:
$_{j}\ \frac{1}{N}\sum_{n=1}^{N}{}^{j}_{e}\ {}^{j2}_{c}$
$_{j}\ \frac{1}{N}\sum_{n=1}^{N}{}^{j}_{out}\ \hat{}$
Gradient Computation and Back-propagation:
Compute gradient $^{J}_{SE}\ \frac{\partial MSE_J}{\partial \theta SE_J}$ and back-propagate it.
Compute gradient $^{J}_{SR}\ \frac{\partial NLL_J}{\partial \theta SR_J}$ and back-propagate it.
Parameter update:
$^{j}_{SE}\ {}^{j}_{SE}\ {}^{j}_{SE}\ {}^{j}_{SR}$
$^{j}_{SR}\ {}^{j}_{SR}\ {}^{j}_{SR}$
Compute on dev set
if $_{dev}\ {}^{pre}_{dev}$ then
Go to next epoch (step-2)
else
Stop training

$$\theta = \theta - \eta(g + \lambda.g) \quad (12)$$

$$\theta = -\eta * g \quad (13)$$

As mentioned in eq 12, the speech recognition gradient g is back-propagated through the speech enhancement QLSTM network, and it plays a vital role in parameter updataion of θ.

5.0 Experimental Setup and Speech Corpus

The acoustic features for quaternion neural networks are extracted as described in [3]. Then, Pytorch-Kaldi [19] and Kaldi toolkit [15] was used to extract log-Mel filter- bank features with their derivatives. An acoustic quater nion Q(f, t) can be described as:

$$Q(f,t) = e(f,t) + \frac{\partial e(f,t)}{\partial t}.i + \frac{\partial e(f,t)}{\partial t}.j + \frac{\partial e(f,t)}{\partial t}.k \quad (14)$$

Here, f denotes the frequency and t denotes the timeframe. The term *e(f, t)* denotes the energy. The Hindi speech dataset [21] was used for joint training. The noise was added into the clean corpora to generate noisy speech features. The label for the QLSTM speech enhancement (SE) module is the original clean speech feature. youtdenotes the label for speech recognition QLSTM module, which were derived from the Kaldi toolkit using forced alignment procedure. In this work, the QLSTM model with three recurrent layers was used in both SE and SR modules. The number of neurons in each layer was fixed to 256. All layers use the tanh activation function. The system is optimized with Adam optimizer. The model is trained for 24 epoch, and decoding is done by the Kaldi toolkit. The joint-training ASR system was implemented with the Pytorch-Kaldi toolkit coupled with the Kaldi toolkit.

Table 1: Hindi Corpus Details

Dataset	Sentences	Duration
Train	800	2.1 hours
Dev	100	10 Mins.
Test	100	10 Mins.

In this work, the Hindi speech dataset was developed by TIFR, Mumbai. The total duration of this dataset is 2.5 hours. It contains the utterances uttered by 100 speakers. It also covers all the phonemes of the Hindi language. It was recorded in a quiet room on 16 kHz sampling frequency. The train, test, and dev set details can be found in Table 1. In the train set, speech utterances were contaminated with babble noise from the NOISEX database. The Dev and Test set are also prepared in the same fashion.

6.0 Results

6.1 Role of i-vector adaptation

In this experiment, we investigate the role of acoustic modeling and i-vector adaptation in

noisy speech recognition. The 40-dimensional FBANK features augmented with and without 100-dimensional i-vector features are the input to all the neural network architectures. This table clearly shows that the QLSTM model performs well in both clean and noisy conditions.

Table 2: Role of i-vector Adaptation

Acoustic Models	without i-vector		with i-vector	
	Clean	Noisy	Clean	Noisy
DNN-HMM	22.50	23.40	20.10	21.40
LSTM	18.60	19.50	17.20	18.40
GRU	16.90	17.80	15.60	16.50
LiGRU	16.20	17.30	14.80	15.90
CNN	14.60	15.50	13.20	14.30
QLSTM	11.80	12.90	10.50	11.40

We observed a 10% WER reduction in the QLSTM model compared to the DNN-HMM model. The i-vector features further reduce the WER 1.5% in all neural network acoustic models. The best WER 10.5% and 11.4% were reported by the QLSTM model in clean and noisy conditions. For DNNHMM, five MLP layers with 1024 hidden neurons in each layer were used. All other neural networks had three hidden layers, with 1024 hidden neurons except the QLSTM model. In the case of the QLSTM model, we used 256 hidden neurons in each layer.

6.2 Performance analysis of joint-training framework

In Table 3, four different strategies were compared on different SNR levels. In the first approach, a single five- layered QLSTM model with 512 neurons in each layer was trained without speech enhancement. In the second approach, three layered QLSTM speech enhancement modules with 256 hidden neurons in each layer were trained.

Table 3: Performance Analysis of Joint-Training Approach with Other Approaches

Approach	LM	Model	20dB	15dB	10dB	5dB	0dB	Avg.
Single Big DNN	3G	QL-	22.60	28.90	34.30	38.40	45.40	33.92
	3G+4G	STM +	22.10	28.40	33.90	38.00	44.90	33.46
	RNN	i-vector	21.80	28.10	33.50	37.60	44.50	33.10
SE+ clear SR	3G	QL-	15.50	21.90	28.20	32.50	38.20	27.26
	3G+4G	STM +	15.1	21.40	27.80	32.10	37.80	26.84
	RNN	i-vector	14.70	21.10	27.40	31.80	37.50	26.50
SE+ matched SR	3G	QL-	14.1	20.20	26.10	30.5	36.40	25.46
	3G+4G	STM +	13.7	19.80	25.80	30.10	36.00	25.08
	RNN	i-vector	13.40	19.40	25.50	29.80	35.70	24.76
SE + SR joint training	3G	QL-	12.40	18.50	24.80	28.50	33.20	23.48
	3G+4G	STM +	12.10	18.10	24.30	28.10	32.80	23.08
	RNN	i-vector	11.80	17.70	23.90	27.80	32.50	22.74

Table 4: Performance (%WER) comparison with other studies independently first, and then this trained model is coupled with a speech recognition module prepared with clean speech. In this approach, the speech enhancement module distorts the speech signal, which is hard to recognize by the speech recognition module trained with clean speech utterances. This problem is addressed by *SE + matchedSR* approach. In this approach, the first speech enhancement was done, and the speech recognition module is trained with enhanced acoustic features. The last one reports the WER achieved by the joint- training approach. Table 3 clearly shows the effectiveness of the joint-training approach. The joint-training frame- work with RNN language model helps to get a relative 2% improvement over matched SR approach.

6.3 Performance (WER%) comparison with other studies

Table 4 compares the results of the proposed system with several other studies on the same dataset. Dua et al. [10] utilize the Differential Evolution (DE) refined Gammatone Frequency Cepstral Coefficient (GFCC) features trained with Maximum Mutual Information (MMI) dis- criminative training for noise-robust Hindi ASR. Dua et al. [9] use a Basilar-membrane Frequency band Cepstral Coefficient (BFCC) features and HMM-based acoustic modeling to get the noise-robust speech features. He further introduces DE optimized BFCC features to enhance system performance. As shown in Table 4, our jointly trained ASR system with the QLSTM model pushed the performance to 22.74% average WER.

Table 4: Performance (%WER) Comparison with Other Studies

Study	Front-end	Back-end	20dB	15dB	10dB	5dB	0db	Avg.
Dua et al.[9]	BFCC	HMM	23.44	26.04	30.6	33.98	49.14	32.64
Dua et al. [10]	DE+GFCC	MMI	14.70	19.70	22.70	31.00	37.00	25.02
Proposed work	FBANK+ i-vector	QLSTM	11.80	17.70	23.90	27.80	32.50	22.74

7.0 Conclusion

This paper describes the impact of the joint-training framework with the QLSTM neural network architecture in noise-robust Hindi ASR. Our proposed QLSTM based joint-training framework has shown significant performance improvement over previously published work for the same dataset. The finding of this work can be summarized as follow:

- The proposed QLSTM model helps to achieve a 2% improvement over other SOTA acoustic models.
- The i-vector adaptation reports the 1.5% relative improvement.
- The proposed RNN language model further reduces WER upto 0.5%.

References

1. M. Brandstein and D. Ward, Microphone arrays, 2002.
2. M. Dua, R. K. Aggarwal and M. Biswas, Performance evaluation of Hindi speech recognition system using optimized filter banks, Engineering Science and Technology, an International Journal, 21 (2018), 389–398.

3. M. Dua, R. K. Aggarwal and M. Biswas, GFCC based discriminatively trained noise robust continuous ASR system for Hindi language, Journal of Ambient Intelligence and Humanized Computing, 10 (2019), 2301–2314.
4. T. Gao, J. Du, L.-R. Dai and C.-H. Lee, Joint training of front-end and back-end deep neural networks for robust speech recognition, in 2015 IEEE Inter- national Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2015, 4375–4379.
5. F. Ge, K. Li, B. Wu, S. M. Siniscalchi, Y. Yan and C.-H. Lee, Joint training of multi-channel-condition dereverberation and acoustic modeling of microphone array speech for robust distant speech recognition, in Interspeech, 2017, 3847–3851.
6. I. Goodfellow, Y. Bengio and A. Courville, Deep learning, MIT press, 2016.
7. E. Hänsler and G. Schmidt, Speech and audio processing in adverse environments, Springer Science & Business Media, 2008.
8. J. Hu and J. Wang, Global stability of complex- valued recurrent neural networks with time-delays, IEEE Transactions on Neural Networks and Learning Systems, 23 (2012), 853–865.
9. S. Kriman, S. Beliaev, B. Ginsburg, J. Huang, O. Kuchaiev, V. Lavrukhin, R. Leary, J. Li and Y. Zhang, Quartznet: Deep automatic speech recognition with 1d time-channel separable convolutions, in ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2020, 6124–6128.
10. S. Makino, T.-W. Lee and H. Sawada, Blind speech separation, vol. 615, Springer, 2007.
11. L. R. Medsker and L. Jain, Recurrent neural net- works, Design and Applications, 5.
12. T. Parcollet, M. Morchid and G. Linarès, A survey of quaternion neural networks, Artificial Intelligence Review, 53 (2020), 2957–2982.
13. T. Parcollet, M. Morchid, G. Linarès and R. De Mori, Bidirectional quaternion long short- term memory recurrent neural networks for speech recognition, in ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2019, 8519–8523.
14. T. Parcollet, M. Ravanelli, M. Morchid, G. Linarès, C. Trabelsi, R. De Mori and Y. Bengio, Quaternion recurrent neural networks, arXiv preprint arXiv:1806.04418.
15. D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz et al., The kaldi speech recognition toolkit, in IEEE 2011 workshop on automatic speech recognition and understanding, CONF, IEEE Signal Processing Society, 2011.
16. M. Ravanelli and Y. Bengio, Interpretable convolutional filters with sincnet, arXiv preprint arXiv:1811.09725.
17. M. Ravanelli, P. Brakel, M. Omologo and Y. Ben- gio, Batch-normalized joint training for dnn-based distant speech recognition, in 2016 IEEE Spoken Language Technology Workshop (SLT), IEEE, 2016, 28–34.
18. M. Ravanelli, P. Brakel, M. Omologo and Y. Bengio, Light gated recurrent units for speech recognition, IEEE Transactions on Emerging Topics in Computational Intelligence, 2 (2018), 92–102.
19. M. Ravanelli, T. Parcollet and Y. Bengio, The pytorch-kaldi speech recognition toolkit, in ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing

(ICASSP), IEEE, 2019, 6465–6469.
20. V. Roger, J. Farinas and J. Pinquier, Deep neural networks for automatic speech processing: A survey from large corpora to limited data, arXiv preprint arXiv:2003.04241.
21. K. Samudravijaya, P. Rao and S. Agrawal, Hindi speech database, in Sixth International Conference on Spoken Language Processing, 2000.
22. Y. Shangguan, J. Li, L. Qiao, R. Alvarez and I. McGraw, Optimizing speech recognition for the edge, arXiv preprint arXiv:1909.12408.
23. J. Song and Y. Yam, Complex recurrent neural net- work for computing the inverse and pseudo-inverse of the complex matrix, Applied mathematics and computation, 93 (1998), 195–205.
24. Z.-Q. Wang and D. Wang, A joint training framework for robust automatic speech recognition, IEEE/ACM Transactions on Audio, Speech, and Language Processing, 24 (2016), 796–806.
25. F. Weninger, H. Erdogan, S. Watanabe, E. Vincent,
26. J. Le Roux, J. R. Hershey and B. Schuller, Speech enhancement with LSTM recurrent neural networks and its application to noise-robust ASR, in International Conference on Latent Variable Analysis and Signal Separation, Springer, 2015, 91–99.
27. D. Yu and L. Deng, Automatic Speech Recognition, Springer, 2016.